JPL SP 43-31

LUMIS INTERACTIVE GRAPHICS OPERATING INSTRUCTIONS AND SYSTEM SPECIFICATIONS

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91103

(NASA-CR-148764) LUMIS INTERACTIVE GRAPHICS OPERATING INSTRUCTIONS AND SYSTEM SPECIFICATIONS (Jet Propulsion Lab.) 101 p N76-30821
CSCL 09B Unclas
G3/60 50450

REPRODUCED BY
NATIONAL TECHNICAL INFORMATION SERVICE
U. S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

Prepared for
Office of Technology Utilization
and
Office of Applications
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

JPL SP 43-31

LUMIS INTERACTIVE GRAPHICS OPERATING INSTRUCTIONS AND SYSTEM SPECIFICATIONS

Nevin A. Bryant
Project Manager

Tong C. Yu
System Architect

Albert J. Landini
City Planner
Los Angeles, California

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91103

August 15, 1976

Prepared for
Office of Technology Utilization
and
Office of Applications
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

PREFACE

This work was sponsored by the Office of Technology Utilization of the National Aeronautics and Space Administration through Contract No. NAS 7-100.

This document is one of two principal systems specifications documents to be generated by the LUMIS program. While this document addresses the problem of data base interrogation, the other[1] presents software and procedures involved in data base construction using the census DIME file and point-in-polygon architectures.

[1] Bryant, N., Paul, C. K., Landini, A. J., Bannester, R. W., and Logan, T., LUMIS Land Use Management and Information Systems Coordinate Oriented Program Documentation, SP 43-33. Jet Propulsion Laboratory, Pasadena, California (to be published).

Preceding page blank

ACKNOWLEDGEMENTS

Many individuals have contributed towards the creation of the system described in the following pages. Dr. Charles K. Paul was responsible for enlisting NASA support for the Land Use Management Information System (LUMIS) from the Office of Applications, and mapping out the overall LUMIS program. The interactive version of LUMIS was conceptualized and implemented on a test case for a portion of the Santa Monica Mountains in Los Angeles by Carl Diegert. Tong C. Yu has been responsible for redesigning the initial system to assure its transportability and operating efficiency.

CONTENTS

CONTENTS (contd)

FIGURES

ABSTRACT

The LUMIS program has designed an integrated geographic information system to assist program managers and planning groups in metropolitan regions. Described is the system designed to interactively interrogate a data base, display graphically a portion of the region enclosed in the data base, and perform cross-tabulations of variables within each city block, block group, or census tract. The System is designed to interface with U. S. Census DIME file technology, but can accept alternative districting conventions.

The System is described on three levels: (1) introduction to the System's concept and potential applications, (2) the method of operating the System on an interactive terminal, and (3) a detailed system specification for computer facility personnel.

PART I

INTRODUCTION

1.0 BACKGROUND

The Land Use Management Information System (LUMIS) was developed by the Jet Propulsion Laboratory (JPL) in cooperation with the Los Angeles City Planning Department for the City portion of the Santa Monica Mountains and with the City of Tacoma Planning Department. LUMIS incorporates data developed from maps and aerial photos as well as traditional land based data associated with routine city and county record keeping activities and traditional census data.

To achieve the merging of natural resource data with governmental data, LUMIS was designed in accordance with restrictions associated with other land use information systems currently being constructed by Los Angeles and Tacoma city staffs. The city systems are based on geographical and environmental data and utilize U.S. Census Bureau DIME file technology. They incorporate numerous local, regional, and Federal data files at the first level of urban geographic aggregation -- the individual census block.

In its interactive mode, LUMIS uses a graphics terminal to display urban street networks. Users can query their data bases and display numerical data values for each of the city blocks shown on the TV-like screen. The interactive system is most efficient at producing special purpose maps for areas consisting of up to one tract at the block level, and correspondingly larger areas at the block group and tract levels. The data base generally consists of block level census data and data digitized from maps and photographs. Census Bureau DIME files provide the basic street network information and a means of aggregating local government data to the block level.

In using LUMIS, urban planners are confronted with a series of "plain English" instructions and commands that guide them through the System. In this sense the interactive graphics portion of LUMIS is self-teaching for users. LUMIS provides commands for study area selection, geographical detail specification, street network mapping, data selection, and data mapping.

LUMIS was developed in such a manner as to make it transferable to the over 200 major cities in the United States having DIME files. As DIME technology becomes more widely accepted internationally, it is likely that LUMIS will also find its share of overseas users.

An objective of the LUMIS task was to demonstrate the potential of interactive computer graphics to urban users to illustrate the value of an advanced LUMIS operating system. For the first time, it is possible for an analyst to create special purpose maps from selected parts of a metropolis, and incorporate the product in a variety of planning activities.

The interactive graphics system basically takes as input, digitized block boundaries and map polygons. The graphics routines display the digitized block polygons in map form and the socioeconomic data in numeric form at the polygon or block centroids. The block boundary map file may be generated by manual digitization or, in most cases, by applying a series of programs to generate a file from the U.S. Census DIME file for that city. The block boundary map file (picture file) interfaces directly with the Third Count Block Census file tapes, and the user has the option available to merge other data sets aggregated at the block level and encoded in the census block, block group, and tract designations.

To date, the interactive graphics system as maintained for the City of Los Angeles encompasses one hundred and seventy-three socioeconomic data items regarding population, housing, and land use for six census tracts having approximately two hundred census blocks in Woodland Hills, California. The socioeconomic data items were obtained by merging the Third Count U.S. Census tapes and the Los Angeles County Assessor Secured File for these six tracts. The City of Tacoma graphics system consists of 55 variables from the 1970 Block Census and will include an additional 70 to 100 variables from the Tacoma Land Use and Building Structure files.

The LUMIS Interactive Graphics Terminal is a Tektronix 4013/4015 utilizing the APL programming language for graphics representation of geographic and tabular data. APL functions have also been written which permit the planner to converse with the computer in English language statements.

2.0 CONCEPTS

The Land Use Management Information System was designed to have two modes of output. The first of these is the traditional computer printer-paper formatted output. On these traditional reports the land referenced activity data would be a series of columns across the page. Each column would be headed with such titles as residential land use and the absolute acreage of that land use; its proportion would be stated as a percentage of the total land area in the reference polygon. Each row in the LUMIS report would be unique for one of the reference polygons that comprised the geographical subdivisions of the entire study area for which the the LUMIS had been instituted. All columns for each row in a LUMIS report would total the complete area for the reference polygon, or 100%.

The second LUMIS mode involves interactive graphics. In addition to the data retrieval powers of the LUMIS traditional reporting system, the interactive graphics mode provides the user with a rudimentary but planimetrically correct streetwork map of the area for which georeferenced data was requested. LUMIS interactive graphics were developed to provide spatially oriented data users with the ability to "see where the data they had requested was located". While the geographic display of data is often times deemed trivial by the non-spatially oriented researcher, he soon learns to respect the power of doing it automatically after completing his first map coloring exercise.

In designing the LUMIS interactive graphics and traditional reporting system serious considerations had to be given to the size and type of geographic unit the data was to be reported for. It was obvious to the LUMIS design staff that until now remote sensing had played a less than optimum role in planning urban centers and that its potential to monitor urban dynamics had not been fully exploited.

In pursuing the LUMIS design it was learned that the processing of imagery, either by interpretation of aerial photos or machine processing of digital imagery, was expensive by any municipal government's budgetary considerations. It was also learned that remote sensing alone could not gather all the information that urban researchers associated with land use data in any way let alone a cost effective one. Socioeconomic and demographic characteristics such as rent rates or age of families could not be supplied by interpreted imagery at any scale.

Thus the LUMIS designers had to consider the types of data that urban planners and researchers needed and used. What was learned is that some types of data were best gathered through the use of traditional field methods such as the decennial census or the processing of routine administrative records from building, assessor and welfare offices.

LUMIS then was developed giving considerations to the cost/benefit comparisons between remote sensing and traditional ground survey of land use and socioeconomic measures. These were weighed against a third dimension of resolution and scale and the question asked, "At what level of geographic detail can the cost ratio of obtaining data from remote sensing be held to a minimum when compared to ground survey data collection."

Another way of looking at the cost question was to first understand that for more detail an aircraft must fly at a lower altitude. When flown at a lower altitude the aircraft must make more passes to cover a given study area. When the aircraft is flown lower and longer the associated costs go up. The LUMIS designers had to consider what they believed was the optimum scale of photography or the altitude the aircraft was to be flown at to produce photos for the LUMIS.

To make this decision they investigated the needs and uses of data by urban planners and researchers both as to type of data and the level of geographic detail at which it was used. They found that remote sensing would not answer all data needs and that beyond a certain level of detail it was not cost effective to use aerial imagery and that in fact extreme low altitude or large scale imagery produced data that was redundant or superfluous to that collected through routine administrative procedures.

The LUMIS design task posed the additional problem of having to overcome the format in which the raw or processed image data was traditionally presented. Urban planners and decision makers think of their areas as a hierarchy of politically well defined geographic units most often in the shape irregular polygons. These users could not incorporate data reported by large abstract grid cells or mapped irregular geophysical polygons easily into their information using processes.

This meant that LUMIS had to report its data by some geographic unit commonly recognized by the urban user and that this geographic unit must also represent that point in the geographic hierarchy at which image based data could most cost effectively be merged with field survey or administrative record base data. To determine at which point in the political geographic hierarchy LUMIS should intersect for data capture one such hierarchy was studied in detail.

3.0 LOS ANGELES'S GEOGRAPHICAL HIERARCHY

3.1 Parcel Level Information System (LUPAMS)

The City of Los Angeles had developed a parcel level information system. It was named the Land Use Planning and Management System (LUPAMS). The basic record for that system was the individual land ownership parcel. There were approximately 700,000 such parcels in the City. The basic data in that system was collected by the Los Angeles County Assessor as part of his property assessment activities.

To the basic Assessor data, City Planning staff added parcel area by physically planimetering each parcel's mapped outline, and census block and census tract geocodes. That geocoding activity was also a manual one in which map correspondence was determined between Assessor maps and Census maps. The required census codes were then transferred to the Assessor maps and, in a second step, coded for addition to the basic Assessor file. When the addition of parcel area and census geocodes to the Accessor file was completed the resulting file was referred to as the LUPAMS file. That file is processed by City users in two ways. Copies of the file's data for each parcel are placed on microfiche and made available to requesting Departments. The LUPAMS data was also maintained in machine readable form (13 computer tapes) and accessed via ASI-ST a file management proprietary software package.

Even at this basic level in the exemplary hierarchy there existed some confusion as to identifying the basic geographical reference unit. The two smallest geographic units found were the lot and parcel. These existed in parallel but referred to slightly different units of land. Parcel denoted a primary ownership unit while a lot was the basic survey unit in a tract within a subdivision. It was possible for one ownership parcel to consist of one or more subdivision lots or fractions thereof.

Parcel identification was how the County Assessor geocoded or geographically identified the physical location of owned properties. This

system was tied to a set of map books cartographically describing the entire County. Each ownership unit was assigned a map book, page and parcel number. There are approximately 800,000 parcels in the City of Los Angeles.

Overlaying both the lot, block and tract system of the City Engineer as well as the book, page and parcel system of the County Assessor was the City's network of streets and highways. Upon investigation it was estimated that over 95% of the lots or parcels within the City conformed to the City's street system. That is, that with rare exception, lots or parcels <u>did not</u> cross streets and were complete land units within a larger unit (the city block) described by the street system itself.

3.2 Block Level Information System (Geo-BEDS/LUMIS)

Geo-BEDS and LUMIS are both block level geographical information reporting systems. That is to say that all geographically referenced data contained in these systems are reported as units or percent of units for each individual city block. Geo-BEDS stands for Geographically Based Environmental Data System, and LUMIS stands for Land Use Management Information System. The city blocks used in both systems are those that were identified and assigned individual numbers (geocodes) by the U.S. Bureau of the Census, for the 1970 census.

As mentioned earlier, each parcel record in the LUPAMS was assigned the census block number in which it was located. Thus both Geo-BEDS and LUMIS had as their basic data input aggregated County Assessor information. In addition Geo-BEDS and LUMIS contained third count census block level population and housing data.

The concepts behind Geo-BEDS and LUMIS represent the merging and blending of politically referenced and coordinate referenced data types into a uniform system. Geo-BEDS was originally conceived to be a nominal or politically referenced geographic data information system. It was to only contain data aggregated through address matching procedures to the individual city block. Geographic access to the System is through a census map reference atlas that has paginal conformation to the City's official cadastral mapping system. Geo-BEDS data processing was sequential file oriented and was to be done through standardized ASI-ST procedures. In this system there are

23,000 records one for each city block. Coordinates were planned to be added to this file for purposes of computer mapping and point and polygon routines, but not for data collection procedures.

LUMIS was devised as an ordinal coordinate referenced system that would allow natural resource or physical geographical data coming from maps and aerial photographs to be aggregated or reported by coordinately defined geographic units. LUMIS was instituted on a test basis for the City and it was decided to report data from aerial photographs as units or aggregates for individual city blocks.

The LUMIS incorporates traditional low altitude photo interpretation, map model construction, and coordinate digitizing procedures. In the test instance the major polygon to which data was aggregated was the individual city block as specified in Geo-BEDS, and the minor or overlayed polygons were the photo interpreted map models. Both major and minor polygons were computer overlayed and statistics indicating incidence of intersection between major and minor polygons were reported. Because the major polygons were defined by the existing street network their digitizing made possible the development of interactive graphic LUMIS software that would portray individual city blocks on a cathode ray tube terminal. The LUMIS and Geo-BEDS data bases allowed users of the interactive graphics LUMIS to make "instant" work maps suitable for inclusion in certain types of reports or to be used as rough draft instruments for professional cartographers. LUMIS interactive graphics opened the door for planners to use both nominal and ordinal data in a real time simulation mode.

3.3 Census Tract Level Information System (SUM)

The Scientific Urban Matrix (SUM) was developed by the Los Angeles Community Analysis Bureau (CAB) and utilizes the census tract as its geographic reporting unit; the final element in our geographical hierarchy. The basic charge of the CAB was to define, identify and locate urban blight in Los Angeles. To that end they produced the first operating automated geographical reporting system. That system was assigned the acronym of SUM

and its data base consisted of records for the 741 census tracts in the City. For each tract over 300 data items were collected through a variety of procedures and those items were made available to all City agencies.

SUM pointed out dramatically how costs of data collection, storage and manipulation went down as the need for exacting geographic specificity lessened. This system also illustrated the minimum level of detail at which geographic data can prove useful for urban planning purposes. Although parcel, block, and even tract data may be aggregated further upward, <u>the raw data from which those aggregations are made can be no more gross than the individual census tract</u>.

4.0 PLANNING ACTIVITIES UTILIZING LAND USE DATA

Any number of taxonomies can be constructed that describe the various operations of an urban planning agency. These can be narrowed somewhat by limiting them to ones that directly involve large area land use data. For sake of brevity we have further restricted our classification of planning activities to those primary urban studies that act as baseline data that can be incorporated in LUMIS and be used in background reports for actual plans. In our taxonomy there are four major groups of operations: 1) population studies, 2) housing studies, 3) economic studies, and 4) studies of social conditions. We will briefly review these to provide a frame of reference for LUMIS users to incorporate interactive graphics in their planning activities.

4.1 Population Studies

Planning agencies are concerned with two major types of population studies. These are population estimates and population projections or forecasts. Estimates are a best "guess" of the actual number of people <u>now</u> residing within a given area, while projections are a forecast or statement of the future number of people expected to be living in that same area. In one sense population estimates are considered as a type of census update. They usually contain data about the numbers and types of dwelling units in an area as well as the population. There are a wide variety of population and housing estimation methodologies. The ones used are usually selected because of data availability constraints facing the agency making the estimate. In practice

these estimates are made to serve as baseline data for other planning groups wishing to monitor either the relationship of population to land use as expressed in terms of housing, traffic, waste water-sewers, or population to social programs such as those associated with schools, police, fire. In both cases geographic location of the estimated population is critical. For Los Angeles the estimated population and housing numbers are reported each year for each of its 741 census tracts. With the implementation of LUMIS the reporting of those estimates for the 23,000 census blocks now becomes possible.

Population projections are generally made using some variety or modification of a cohort survival model. In making these projections the final numbers are extremely sensitive to birth and death rates. Yet these rates are not the most controversial aspect of population projections. Experts making local population projections most usually disagree on the expected numbers of population migrating into and out of an area.

Trend analysis of land use changes could be an important tool in projecting population migration. Again as with the population estimation numbers Los Angeles reports its population projection numbers at the census tract level of geographic specificity.

In both the population estimation and projection processes, urban land use data annually reported by acres within census polygons would be immeasurably beneficial. Population estimators could universally report housing unit estimates by type and location. This could be done by monitoring changes in land use acreages annually and field checking density acreage relationships for new areas. It is anticipated in the future that population projectors could utilize satellite time series information for each census polygon to determine how land use in that polygon has changed. That data coupled with housing unit vacancy estimations could provide clear insight to urban population migration patterns.

4.2 Housing Studies

We have seen that the counting, estimating and projecting of population is tied directly with similar activities regarding housing. While demographic researchers are seeking to develop techniques for estimating and projecting age, race, and sex distributions and locating them geographically, housing

researchers are seeking to identify housing quality and urban blight and locate those phenomenon geographically.

Annual reporting of urban land use categories by acres, and by census areas will aid housing researchers in routinely specifying "where" in the city various types of residential structures can be found, such types as single family dwellings, two story apartments, and multistory apartments. Interpreted aerial photo data also seems to hold out the promise to be able to geographically classify the location of these areas as "sound" or "unsound", "good" or "bad quality", and "blighted" or "nonblighted". More importantly aerial photography would provide a reliable early warning device through time series monitoring to identify geographical areas going through critical neighborhood change.

Specifically housing planners need to know a variety of things about the existing housing stock. They need to know where it is located by type and quality, how many people live in each quality and structure type, how fast each quality and structure type is being removed from the housing inventory and how fast it is being replaced.

In Los Angeles these questions are not being answered directly, but rather through a series of approximations. Skillful incorporation of photo based data into analytical systems such as LUMIS could provide a great deal more reliable information than is currently available regarding housing numbers and quality and in a manner that would allow Federal agencies to have a more complete and uniform housing "picture" of any given urban area.

4.3 Socioeconomic Studies

Urban planners are concerned with three states of geographical being. These are similar to those commonly dealt with each day by all of us and known as the past, present, and future states of time. The planner's states of being are existing land use (what is currently on the land), existing zoning (what can currently be placed on the land — under law), and planned land use (what advisory agencies believe to be the best and highest use of the land). We have seen that estimated and projected numbers of population and housing are critical tools for urban planning and that they rely on current or historical land use data or surrogates.

But urban plans deal with more than geographically locating numbers of people in a variety of housing types. Those plans consider where people work and shop, the things they buy and make and the routes they travel. Plans go farther than just providing a framework for efficient urban development. They also consider the more humane aspects of living, its joys and fears. Plans locate parks, hospitals, firehouses, and jails. They consider income as well as basic human rights in designing a variety of lifestyles.

All of these considerations are ultimately expressed by planners in maps or map-like diagrams for each of the three states of geographical being. Just as historical research forms the basis of much of man's aspirations and goals for the future, maps of existing land use are the single critical tool upon which planners lean most heavily in drawing their future plans. It is easy to see the circular nature of how existing land use largely determines future plans and how those plans influence future land use. The careful and systematic monitoring of changing land use is all important if the plans are to reach fruition.

Considerable effort and expense is encountered by municipalities in making, implementing and monitoring city-wide land use plans. Reams of statistical data are gathered and impressive computer systems constructed. But these important tools would be considerably more effective if they could be quickly reduced to maps of existing land use, existing zoning and planned land use. Instant hard-copy maps of LUMIS data are just such a tool.

Earlier we discussed nominal and ordinal systems. Digital image data interpretation is clearly an ordinal system. Zoning classification of ownership parcels is clearly a nominal one. The power of each of these two system types is brought into play when data maps of existing land use and existing land zoning with census boundary overlays are placed side by side annually and compared to an existing land use plan.

PART II

THE OPERATING INSTRUCTIONS

All users of the LUMIS interactive graphics system will have to prepare their own basic set of instructions that will direct users in "how to turn the machine on," and "how to get the LUMIS up." It is suggested that these basic instructions be placed permanently at the terminal itself. A sample instruction sheet as those used by the City of Los Angeles is shown in Fig. 2-1.

Once the user has worked through the system initialization instructions, he may type the word "HELP." Fig. 2-2 illustrates the terminal response to the "HELP" command by listing the various modules available to the user. Each module may be accessed by its command statement. Fig. 2-3 illustrates the kinds of response expected from technical queries under various modules available to the user.

To aid beginning LUMIS users a series of illustrative problems have been worked using the major system commands (see Figs. 2-4 through 2-11). A single replication of each of these problems is usually sufficient to acquaint the user with each of the modules.

On occasion the user may become overenthusiastic in his attempt to derive maximum information from a particular application, with the result that he may overload the work space in the computer. Each computer system has its own limit, and its own error message notifies that the workspace is full. It is advised that the user experiment with the system to see the limits of the application, and adjust his operations accordingly (e.g., making several maps and piecing them together later).

Persons operating the system will also learn that much more economical processing can be achieved if they outline the steps to be taken prior to sitting at the terminal, and bring along a metropolitan map and table of variable descriptions.

How to Initiate LUMIS Interactive Graphics System

1. Bottom Rear Panel Switch Settings
   a. Baud Rate
      1) Transmit = 300
      2) Receive = 300
   b. Duplex
      1) Full duplex local copy
   c. Carriage Return
      1) LF (line feed)
2. Set Keyboard Switches to

   LINE and APL
3. Turn Power Switch On (below keyboard on front panel)
4. Make sure signal cable is connected to general design data transceiver from terminal.
5. Push talk button on grey "Data-phone" and dial the computer installation connect number (x-xxx-xxx-xxxx).
6. Upon hearing the high-pitched "data-tone" in the "Data-phone" push the "originate" button on the General Design Data Transceiver and place the "Data-phone" receiver in the appropriate transceiver couplings.
7. Press carriage return.

[Follow the sign-on procedure outlined by the computer service company being used.]

Fig. 2-1. Sample instructions

HELP TYPE_THIS	TO_DO_THIS
COST	DISPLAY YOUR CHARGES FOR USE OF THE COMPUTER SINCE YOU SIGNED ON TO THE APL SYSTEM.
DISPLAY	SELECT AREAS AND DATA, AND DISPLAY THEM.
CENTROID	CORRECT THE CENTROID IN THE INDEX FILES.
EDIT	CORRECT THE COORDINATES IN THE PICTURE FILES.
RESET	CLEAR ANALYSIS REGION.
	CLEAR UNKNOWN ERROR CONDITION AND RESTART.

'HELP' IN THE WORK SPACE 'PROCESS'

HELP TYPE_THIS	TO_DO_THIS
BLDIDX	CREATE INDEX FILES 31,32,33.
BLDPIC	CREATE PICTURE FILES 21,22,23,24.
BLDTAB	CREATE TABULAR FILES 11,12,13.

'HELP' IN THE WORK SPACE 'FILE'

Fig. 2-2. HELP

Question	Answer	Response
Tract Number(s)?	1234 CR	Tract 1234 is selected.
	CR	Previous tract is selected again.
	Q CR	Display is terminated.
Group Number(s)?	100 CR	Group 100 is selected.
	CR	Previous tract is selected again.
	Q CR	Display is terminated.
Block Level?	Y CR	Block level data is displayed.
	N CR	Block level data is noi included.
Data?	D1 CR	Data number 1 is selected.
	D1 + D245 CR	Sum of data number 1 and 245 is displayed. The valid operators are: +, -, x and ÷.
	CR	Previous data is displayed.
	I CR	Census Bureau Identification numbers are displayed.
	Q CR	Display is terminated.
Title?		Any number of characters to be displayed as the title of display.
OK?	Y CR	Cause to pass the area displayed.
	N CR	Cause to start edit.
What?	A CR	Add segment.
	D CR	Delete segment.
	Q CR	End of edit.
	R CR	Replace segment.

CR : carriage return

Fig. 2-3. Questions and answers

```
     DISPLAY
GRAPHIC TERMINAL?
Y
MAP?
Y

NEW AREA?
Y
LEVEL?
T
TRACT NUMBER(S)?
607,612,613
DATA?
I

NEW SCALE IS 168.4 FEET/INCH
NEW LOWER LEFT CORNER IS:
1506441 FEET EAST BY
702413 FEET NORTH.
PRESS RETURN KEY TO CONTINUE
```

TRACT: 607 612 613.

CENSUS BUREAU ID

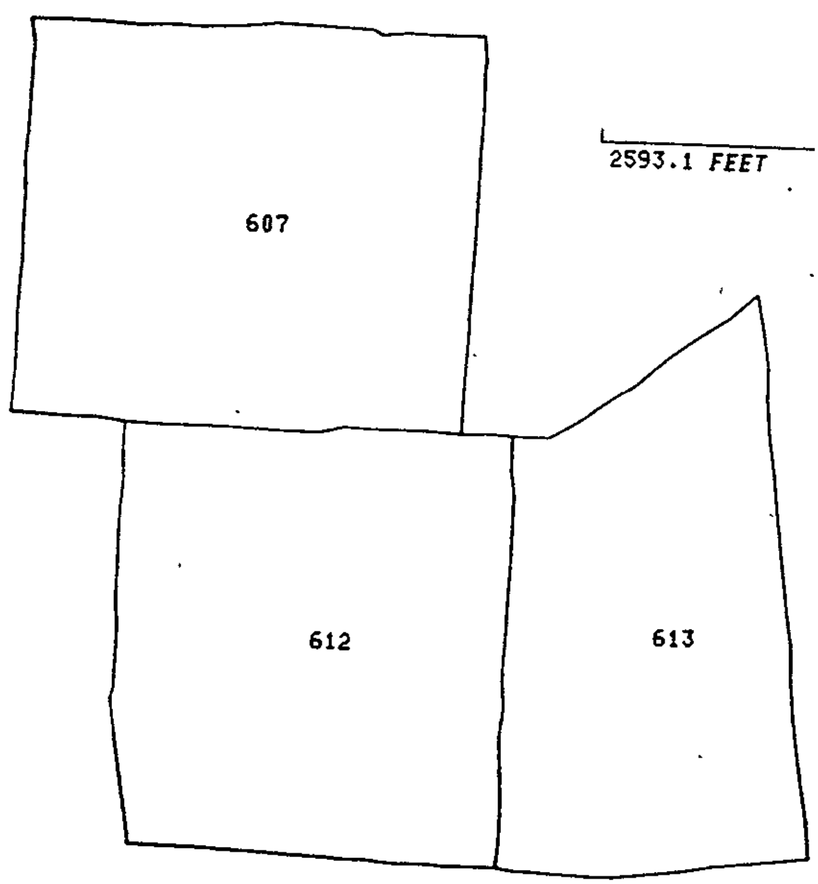


Fig. 2-4. Multiple tract display at tract level

```
NEW AREA?
Y
LEVEL?
G
TRACT AND GROUP NUMBERS?
1: 612,ALL
2: 613,ALL
3:
DATA?
I

NEW SCALE IS 117.4 FEET/INCH
NEW LOWER LEFT CORNER IS:
1507664 FEET EAST BY
702413 FEET NORTH.
PRESS RETURN KEY TO CONTINUE


TRACT  GROUPS
612 100 200 300 400 500 600
613 100 200 300 400 500

CENSUS BUREAU ID
```

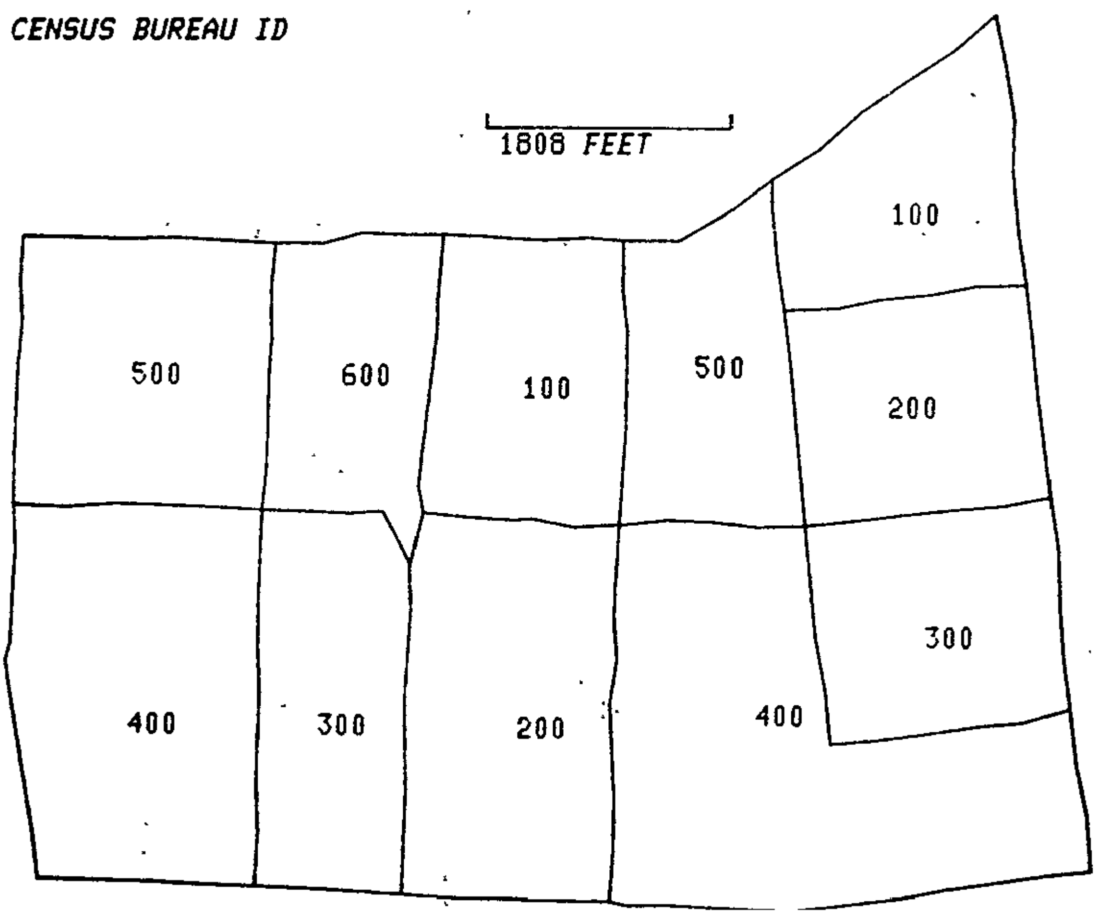


Fig. 2-5. Multiple tract display at block group level

```
NEW AREA?
Y
LEVEL?
G
TRACT AND GROUP NUMBERS?
1: 613,ALL
2:
DATA?
D1+D2
MAXIMUM IS 191
MINIMUM IS 81
TITLE?
OWNED HOUSE AND RENTED HOUSE

NEW SCALE IS 117.4 FEET/INCH
NEW LOWER LEFT CORNER IS:
1512120 FEET EAST BY
702413 FEET NORTH.
PRESS RETURN KEY TO CONTINUE
```

```
TRACT  GROUPS
613 100 200 300 400 500

OWNED HOUSE AND RENTED HOUSE
```

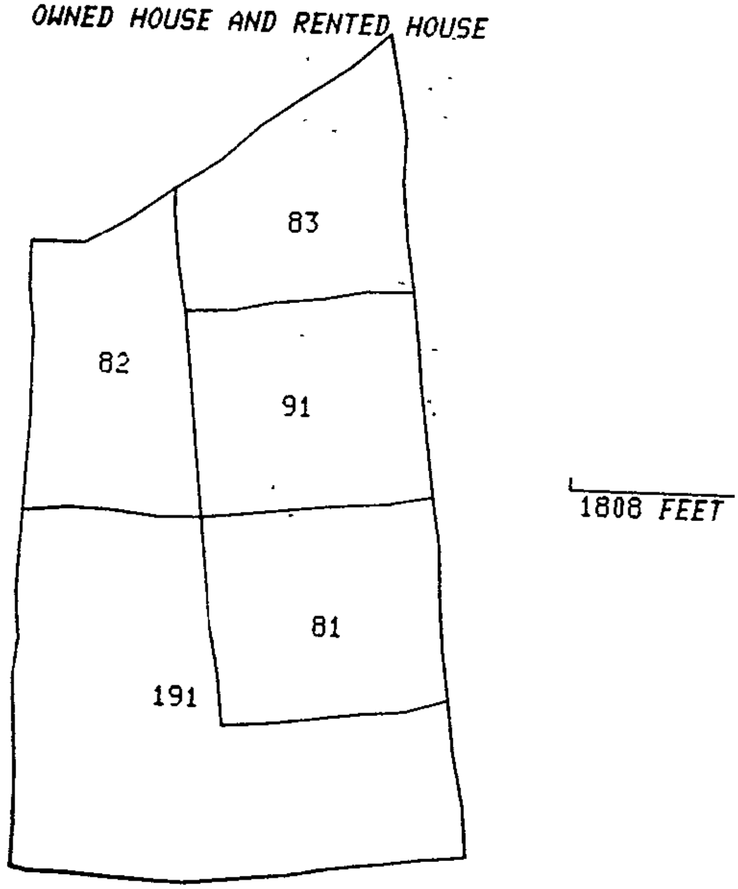


Fig. 2-6. Single tract display at block group level

ORIGINAL PAGE IS
OF POOR QUALITY

```
NEW AREA?
Y
LEVEL?
B
TRACT AND GROUP NUMBERS?
1: 613,ALL
2:
DATA?
D1+D2
MAXIMUM IS 19
MINIMUM IS 0
TITLE?
OWNED HOUSE AND RENTED HOUSE

NEW SCALE IS 117.4 FEET/INCH
NEW LOWER LEFT CORNER IS:
1512120 FEET EAST BY
702413 FEET NORTH.
PRESS RETURN KEY TO CONTINUE

TRACT  GROUPS
613 100 200 300 400 500

OWNED HOUSE AND RENTED HOUSE
```

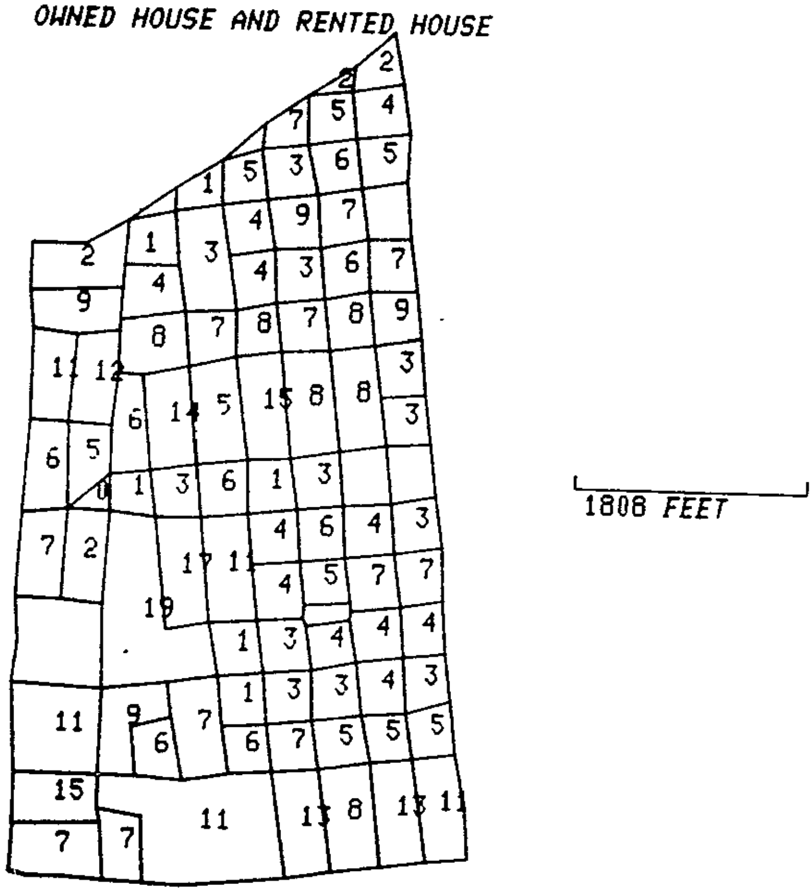


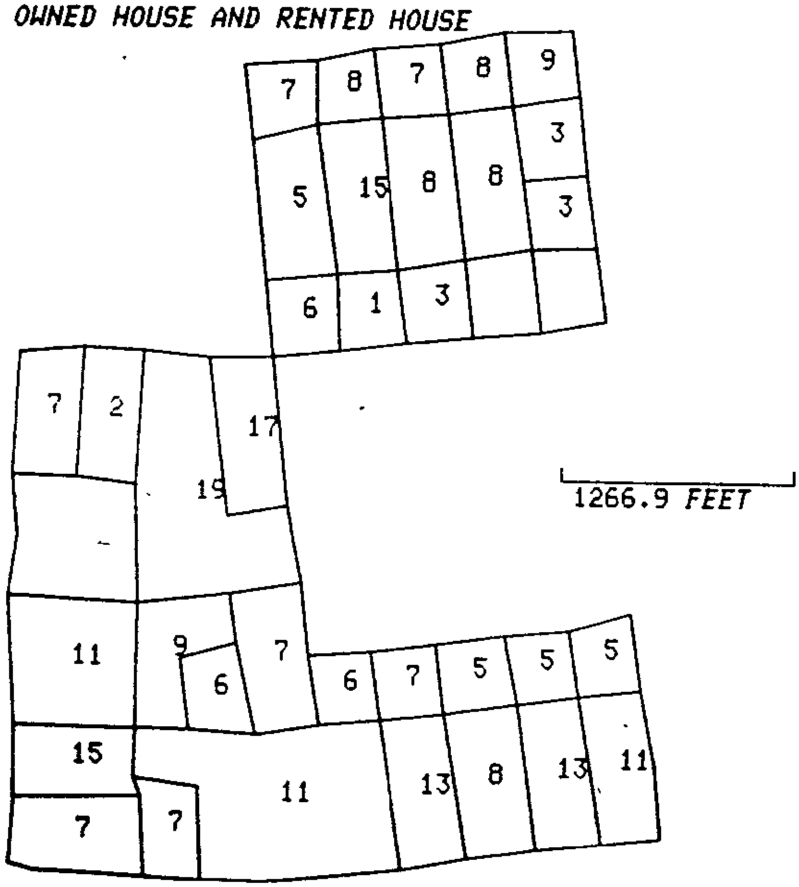
Fig. 2-7. Single tract display at block level

```
NEW AREA?
Y
LEVEL?
B
TRACT AND GROUP NUMBERS?
1: 613,200,400
2:
DATA?
D1+D2
MAXIMUM IS 19
MINIMUM IS 1
TITLE?
OWNED HOUSE AND RENTED HOUSE

NEW SCALE IS 82.3 FEET/INCH
NEW LOWER LEFT CORNER IS:
1512120 FEET EAST BY
702413 FEETNORTH.
PRESS RETURN KEY TO CONTINUE

TRACT  GROUPS
613 200 400

OWNED HOUSE AND RENTED HOUSE
```



ORIGINAL PAGE IS
OF POOR QUALITY

Fig. 2-8. Selected block groups of single tract displayed at block level with crosstabulated variables

```
NEW AREA?
Y
LEVEL?
G
TRACT AND GROUP NUMBERS?
1: 612,100,200
2: 613,400,500
3:
DATA?
D1+D2
MAXIMUM IS 191
MINIMUM IS 82
TITLE?
OWNED HOUSE AND RENTED HOUSE

NEW SCALE IS 95.5 FEET/INCH
NEW LOWER LEFT CORNER IS:
1510607 FEET EAST BY
702413 FEET NORTH.
PRESS RETURN KEY TO CONTINUE

TRACT  GROUPS
612 100 200
613 400 500

OWNED HOUSE AND RENTED HOUSE
```

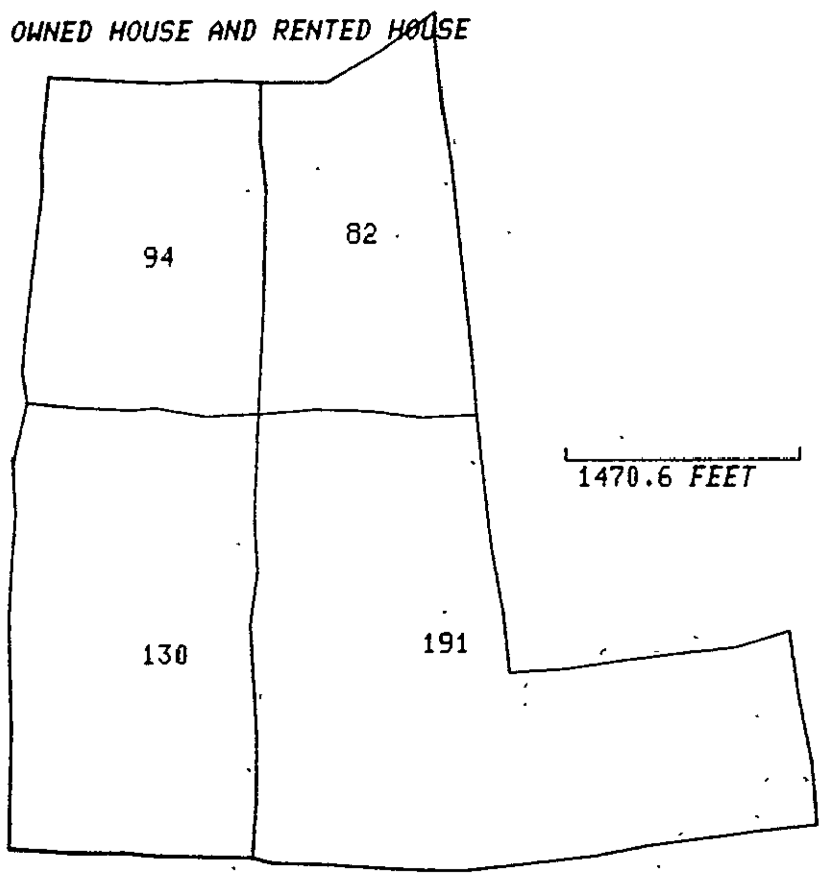


Fig. 2-9. Selected block groups from multiple tracts, displaying group level data

```
NEW AREA?
Y
LEVEL?
B
TRACT AND GROUP NUMBERS?
1: 612,100,200
2: 613,400,500
3:
DATA?
D1+D2
MAXIMUM IS 19
MINIMUM IS 0
TITLE?
OWNED HOUSE AND RENTED HOUSE

NEW SCALE IS 95.5 FEET/INCH
NEW LOWER LEFT CORNER IS:
1510607 FEET EAST BY
702413 FEET NORTH.
PRESS RETURN KEY TO CONTINUE

TRACT  GROUPS
612 100 200
613 400 500

OWNED HOUSE AND RENTED HOUSE
```

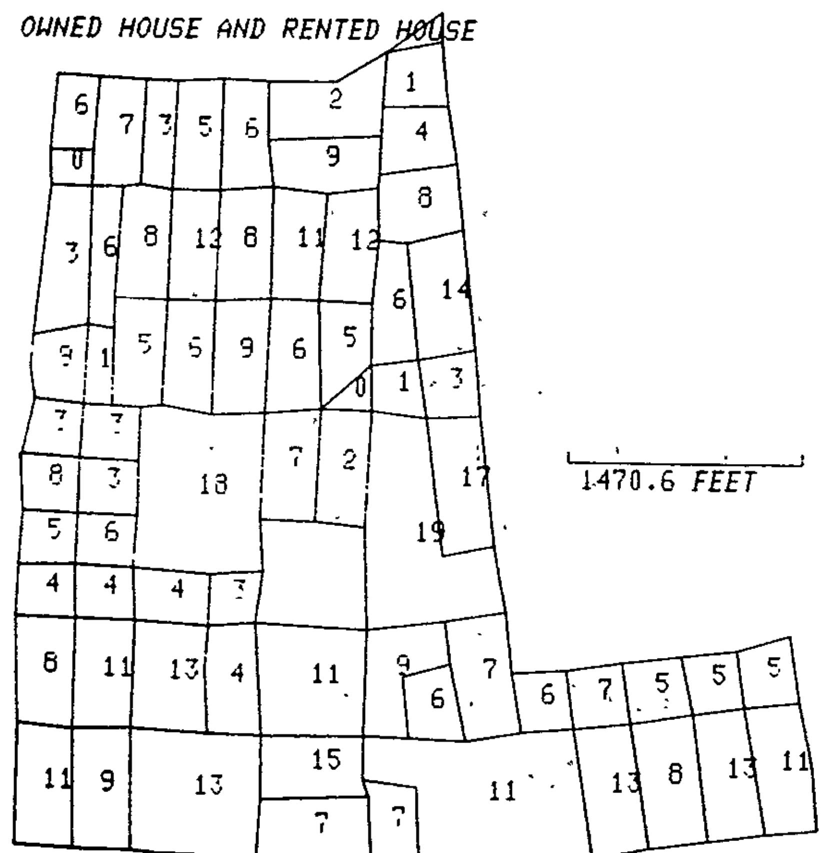


Fig. 2-10. Selected block groups from multiple tracts, displaying block level data

```
     DISPLAY
GPAPHIC TERMINAL?
Y
MAP?
N

NEW AREA?
Y
LEVEL?
G
TRACT AND GROUP NUMBERS?
1: 612,ALL
2: 613,ALL
3:
DATA?
D1+D2
MAXIMUM IS 191
MINIMUM IS 77
TITLE?
OWNED HOUSE AND RENTED HOUSE
PRESS RETURN KEY TO CONTINUE

        TRACT GROUP OWNED HOUSE AND RENTED HOUSE
         612.0   100   94
         612.0   200   130
         612.0   300   91
         612.0   400   77
         612.0   500   105
         612.0   600   88
         613.0   100   83
         613.0   200   91
         613.0   300   81
         613.0   400   191
         613.0   500   82

        NEW AREA?
```

Fig. 2-11. Tabular output of cross-tabulated data

1.0 COMMANDS TO DISPLAY DATA

USER COMMAND	SYSTEM RESPONSE
COPY PROCESS	Copies the APL functions in the work space process into an active APL workspace. This workspace contains the functions called by the LUMIS user to obtain maps and tables of data items or algebraic combinations of data items corresponding to selected tracts, block groups or blocks in the LUMIS data-base. A map or table may contain a mixture of tracts, block groups and blocks. These are simply referred to as <u>areas under analysis</u>. Block groups are always called <u>groups</u>. The PROCESS workspace also contains functions called internally by the functions called by the user.
COST	The cost of using the computer since the user signed-on is displayed.
DISPLAY	This is the major command that the user will have to use for a display purpose. The user answers questions posted by the functions to select areas and data for analysis. The display is terminated when the user answers with a Q when any one of the questions is posted. See Fig. 2-4 for the questions and answers.
HELP	See Fig. 2-2.
RESET	"Clears areas under analysis" by initiating certain vectors and variables in the work-space describing these data.

2.0 COMMANDS TO CREATE DATA

USER COMMAND	SYSTEM RESPONSE
COPY FILE	Copies the APL functions in the work space FILE into an active workspace. This workspace contains the functions called by the user to create/and update data files.
ASSIGN	Assigns input data set name to an APL file name to which an APL function is accessible.

USER COMMAND	SYSTEM RESPONSE
BLDPIC	Creates APL Picture Files 21, 22, 23, 24. See Section 3 of Part IV Program Description for more details.
BLDTAB	Creates APL Tabular Files 11, 12, 13. See Section 3 of Part IV Program Description for more details.
BLDIDX	Creates APL Index Files 31, 32, 33. See Section 3 of Part IV Program Description for more details.
CENTROID	This function is used when an error is detected in the centroids of Index File which is initially created by the function BLDIDX. After defining exactly one block group of block level data under analysis, this function first displays the map of the area then allows one to identify each block. To identify a block, move the crosshairs to the position where the block's data value should be printed, then press the space bar and return. To quit before all are defined, type "Q".
EDIT	After placing exactly one block group of block level data under analysis, this function allows one to add, delete or replace street segments. Type EDIT and follow the instructions on the screen. Block boundaries are edited first and group boundaries are edited next. When the computer asks 'WHAT?', type as follows: A to add — when the cross-hair appears move it to a beginning position of the segment you wish to add and press the space bar first then return. Repeat as often as needed. It requires a minimum of two inputs: from and to positions. Type a Q and return to terminate input. D to delete — when computer asks 'NODE NUMBER?' type one of the numbers that are displayed at each node. Continue the same as many times as needed. Type a Q to terminate. R to replace — (Same as D above.)

3.0 SAMPLE PROBLEMS

With the initial set of commands found listed under HELP most users will find sufficient computer power to meet their data and mapping needs. LUMIS is largely a self-teaching system, and persevering users will gain considerable skill in use of the system without the aid of direct instruction.

To aid beginning LUMIS users a series of illustrative problems have been worked using the major system commands. A single replication of each of these problems is usually sufficient to acquaint the operator with each of the modules.

PART III

SYSTEM SPECIFICATIONS

1.0 INTRODUCTION

The LUMIS interactive version (LUMIS-IV) is an on-line computer software system which operates on a computer operating system supporting APL and FORTRAN programming language.

The purpose of LUMIS-IV is to provide a dynamic method of retrieving meaningful data and display them instantly on terminal in the form of map or table through an on-line, interactive, computer operating system.

The LUMIS-IV Data Base Files are initially created by applying a series of programs written in FORTRAN programming language taking as input Third Count Summary Tape and Geographic Base File (DIME) of Bureau of the Census. The user has the option available to merge other data sets aggregated at the Block level and encoded in the Data Base File. If necessary, the user can also provide digitized block boundaries and map polygons in the Data Base File. Further manipulations of the Data Base Files as well as the retrieval and display of data on terminal are performed by interactive functions (programs) written in APL.

LUMIS-IV has operated on several Time Sharing Operating Systems. The system can operate on a variety of keyboard terminals in the tabular mode, but was designed for graphical output on the Tectronics 4013 and 4015 terminals.

2.0 SYSTEM DESCRIPTION

2.1 Operating System

LUMIS-IV can be transferred onto any operating system that supports APL and FORTRAN programming languages with a minimum conversion effort. The software is programmed to call only basic APL and FORTRAN IV-G subroutines supported by IBM.

2.2 Terminal

Any terminal with standard APL character keys may be used to operate LUMIS-IV. However, a graphic terminal capable of displaying APL and ASCII characters is required in order to display area maps.

2.3 Language

LUMIS-IV programs are written in APL and FORTRAN IV. In order to minimize conversion efforts of users from one system to another, only the standard version or code of programming languages are used in LUMIS-IV. In this particular version the CTS System Commands of Boeing Computer Services are used to control the operations of LUMIS-IV.

2.4 APL Work Space

LUMIS-IV consists of two work spaces; PROCESS and FILE.

1) Work space PROCESS contains functions to retrieve data, manipulate data and display data. The major functions stored in this work space are: DISPLAY, AREA, DATA, MAP, TABLE, CENTROID, EDIT, COST, RESET.

2) Work space FILE contains functions to create files and update files. The major functions stored in this space are: CRCENT CRIDX , CRPIC, CRTAB.

HIERARCHY OF APL FUNCTIONS

MAIN	SUB	SUB	SUB
CRCENT	CONVC		
CRCENT1	DFILE		
CRCENT2			
CRPIC	CONVP		
	DFILE		
CRTAB	CONVT		
CRTAB2	DFILE		
CRIDX			

HIERARCHY OF APL FUNCTIONS (continued)

MAIN	SUB	SUB	SUB
CR10			
CR20			
CR30			
CR40			
ER10			
ER20			
ER30			
ER40			
MESSAGE			
CENTROID	AREA		
COST	PREP		
DISPLAY	DATA	ANS	
	INIT	TYPE	
		DIV	
		ID	
	MAP	BLK	
		CHAIN	
		GIN	
		GRAPH	BLK
		NODE	
		SCALE	
		MIN	
	AREA	ANS	
	TABLE		
	TIEFILE	TIE10	
		TIE20	
		TIE30	
		TIE40	
EDIT	AREA	DECODE	
RESET		EDIT NODE	

3.0 PROGRAM DESCRIPTION (Note Fig. 3-0)

3.1 APL Programs to Display Data (Note Fig. 3-1)

3.1.1 DISPLAY. DISPLAY is the root function of Work Space PROCESS. initializes global variables and monitors overall process of retrieving and d playing data.

ORIGINAL PAGE IS
OF POOR QUALITY

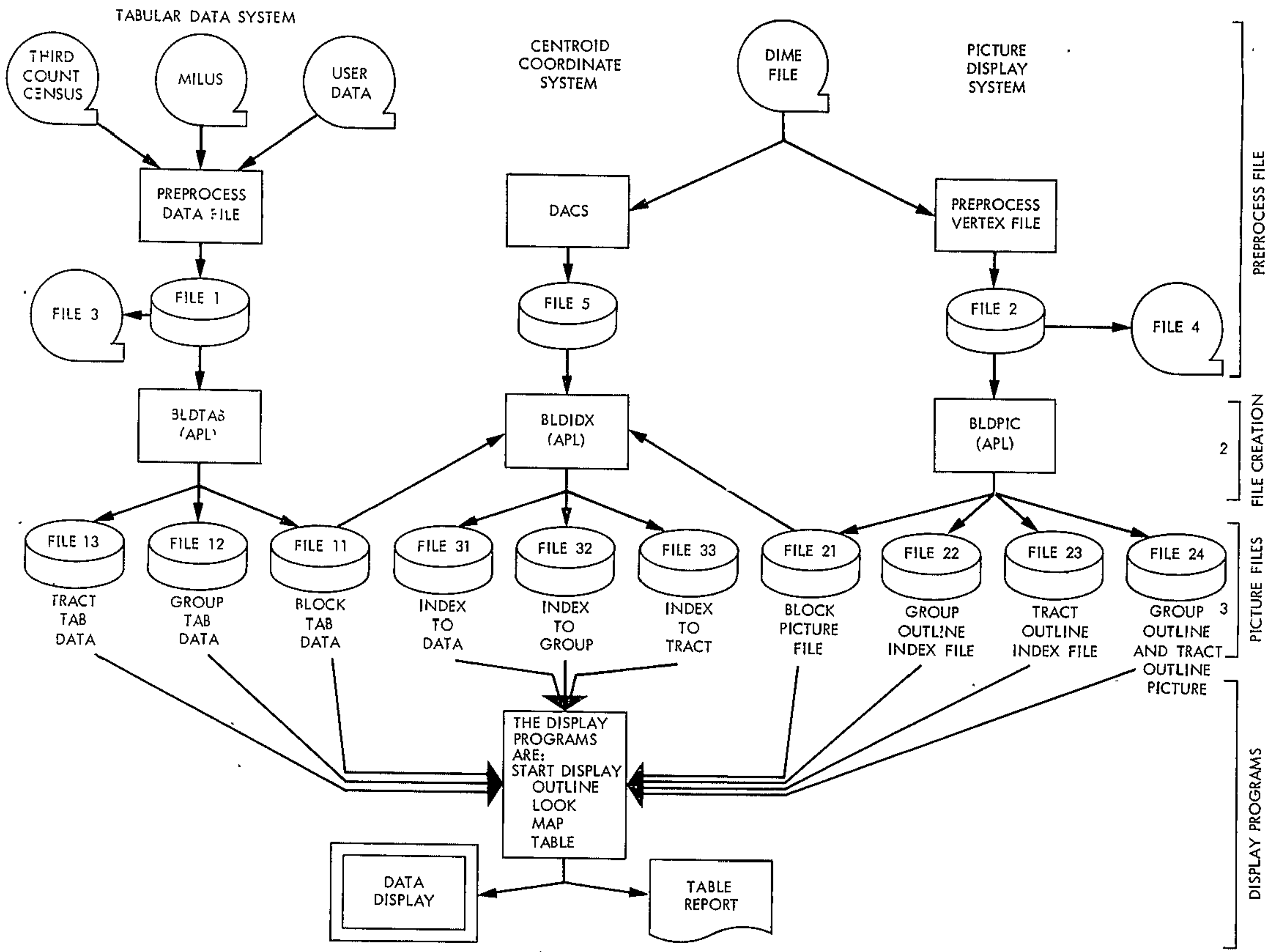


Fig. 3-0. LUMIS-IV Display Flow Diagram

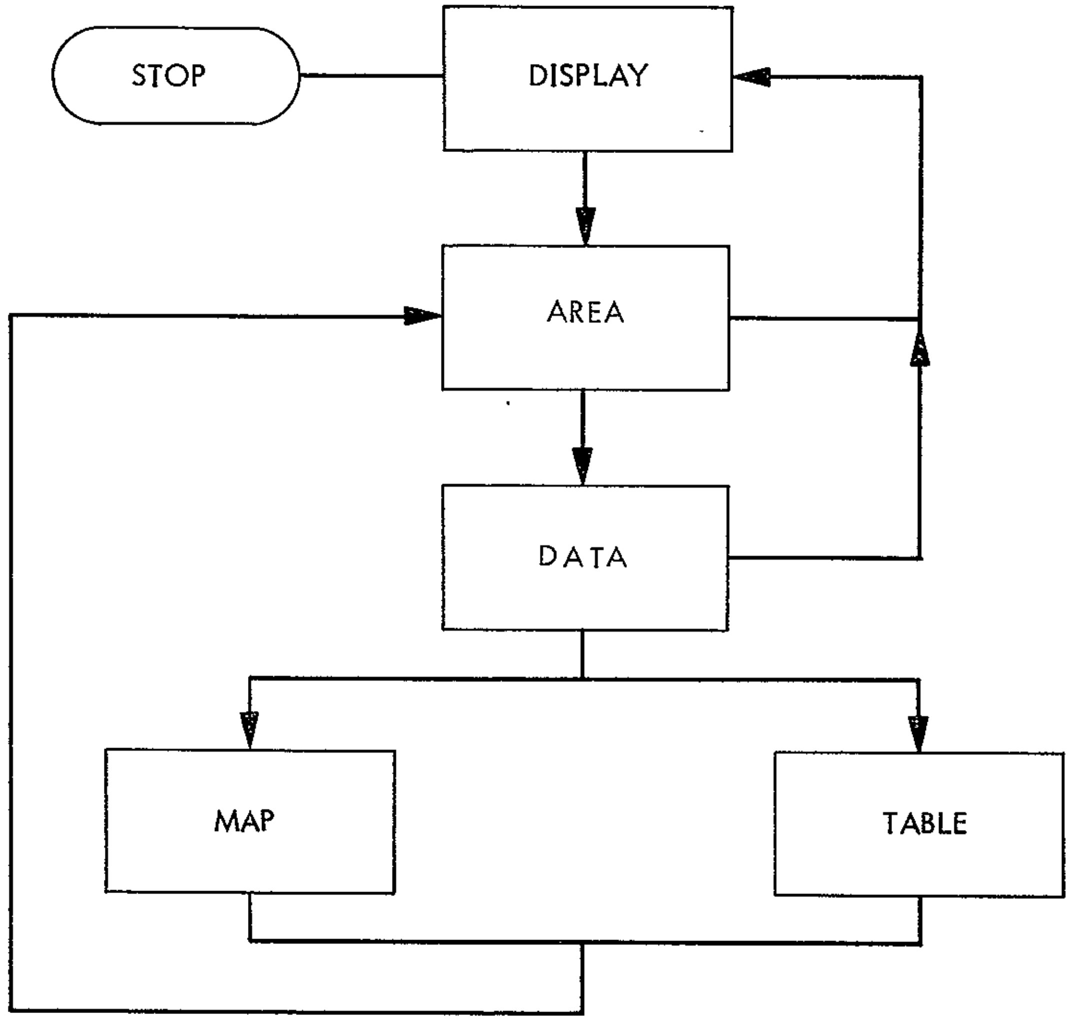


Fig. 3-1. Display Procedure

The Program reads Tract Number from File-30 and retains it as TIF in the memory.

3.1.2 AREA. AREA defines the area to be displayed in terms of Tract, Group and Block.

It generates the Tract Index TIX, Group Index GIX and Block Index BIX.

The user answers questions posed by this function to select areas for analysis. He may call AREA immediately after RESET to initially place areas under analysis. He may also call AREA later to place additional areas under analysis. A reference map (Appendix A) is useful in determining which pieces of census geography are desired for analysis.

3.1.3 DATA. The user answers a question posed by this function to specify one or more data items he would like to use in subsequent maps and tables. DATA retrieves the data items for the areas under analysis from the files. The data items are listed in Appendix II.

3.1.4 MAP. After obtaining from the user a data items, this function draws a street map of the areas under analysis on the terminal, then writes the value of the data item for each area at a preselected point in the block (the point initially selected at the block center).

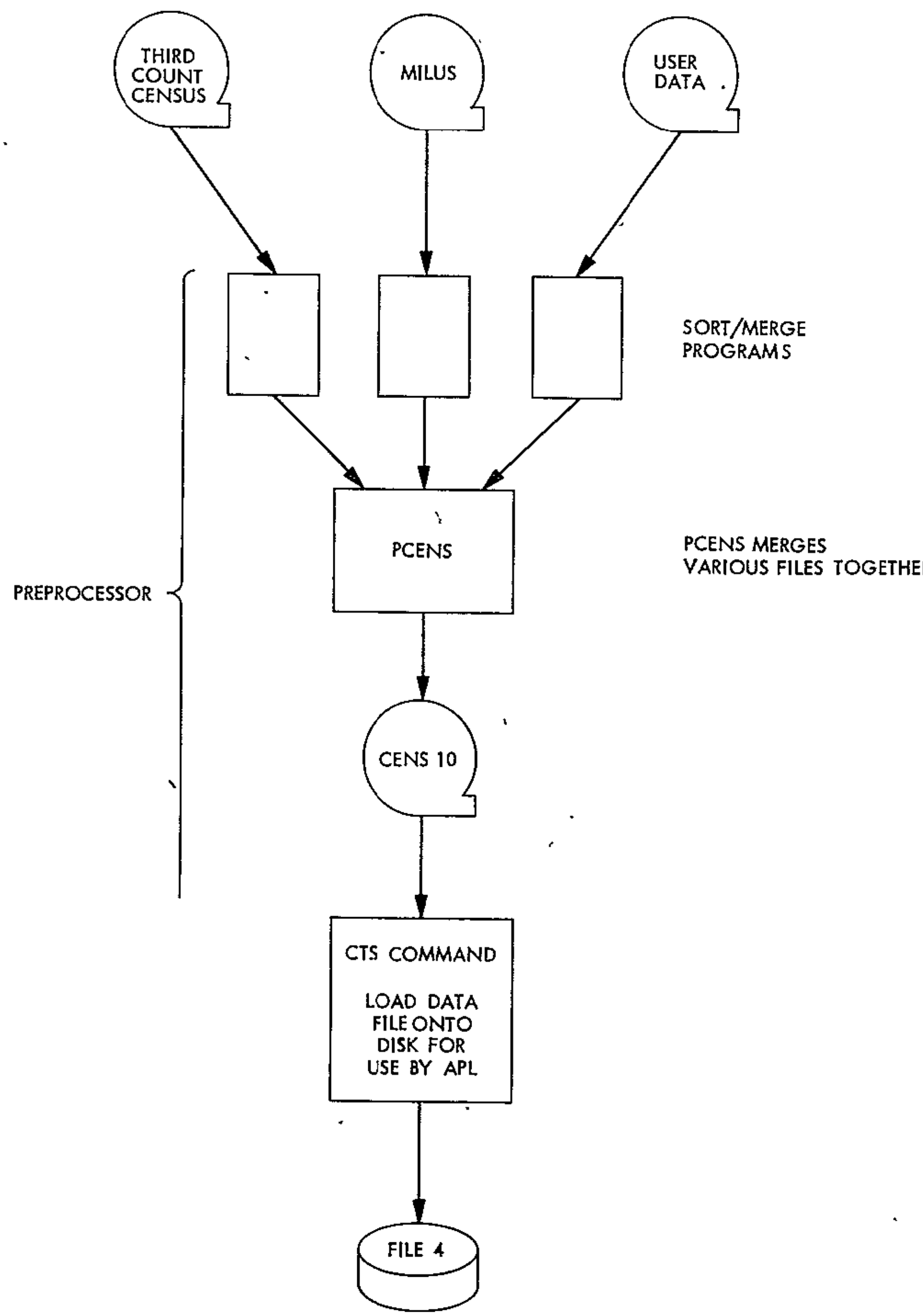
3.1.5 TABLE. Similar to map except a table of the data item(s) with their census identification number is generated. A graphic terminal is not required to use this function.

3.2 Tabular Data File Preprocessor (Note Fig. 3-2)

3.2.1 DATA Sources. DATA sources for the TABULAR DATA BASE can be from a variety of sources; Census, ERTS imagery, polygon overlay or unique files created by an agency for their own use. Files such as land use, licenses or health statistics are common among agencies. Any Census blocks data input can be used by this display technique.

3.2.2 Sort/Merge. Prior to creating a single input file on tape each input file must be merged and sorted by comparable ID codes, forcing each file into a parallel order.

3.2.3 PCENS. PCENS reads data from the Third Count Census Tape and creates a tape file, CENSOO. CENSOO is input to the APL function, BLDTAB. PCENS also reads user provided data, if there is any, and merges it with census data.



ORIGINAL PAGE IS
OF POOR QUALITY

Fig. 3-2. Tabular Data

3.3 CENTROID Coordinate Preprocessor - DACS (DIME Area Centroid System) (Note Fig. 3-3)

3.3.1 DACS - Phase I. DACS is written in two phases. In the first phase DACS creates an abstraction of standard 300 Character DIME files. The abstraction consists of the left and right regions, from and to nodes, and the x and y coordinates. The abstract is written in three levels. Level 1 is for entire Census Tracts. Level 2 is for Block Groups and the third level is Block. Each level is written in contiguous groups on the same temporary file.

3.3.2 System Sort. The DACS file is sorted by the left key or region in ascending order and passed to another temporary file.

3.3.3 DACS - Phase II. The second phase of DACS performs computations, edits the file and writes any output files. The program reads the sorted intermediate tape and stores segment data for a region internally. When the first record for a different region is encountered, the accumulated data is passed to subroutine CHAIN.

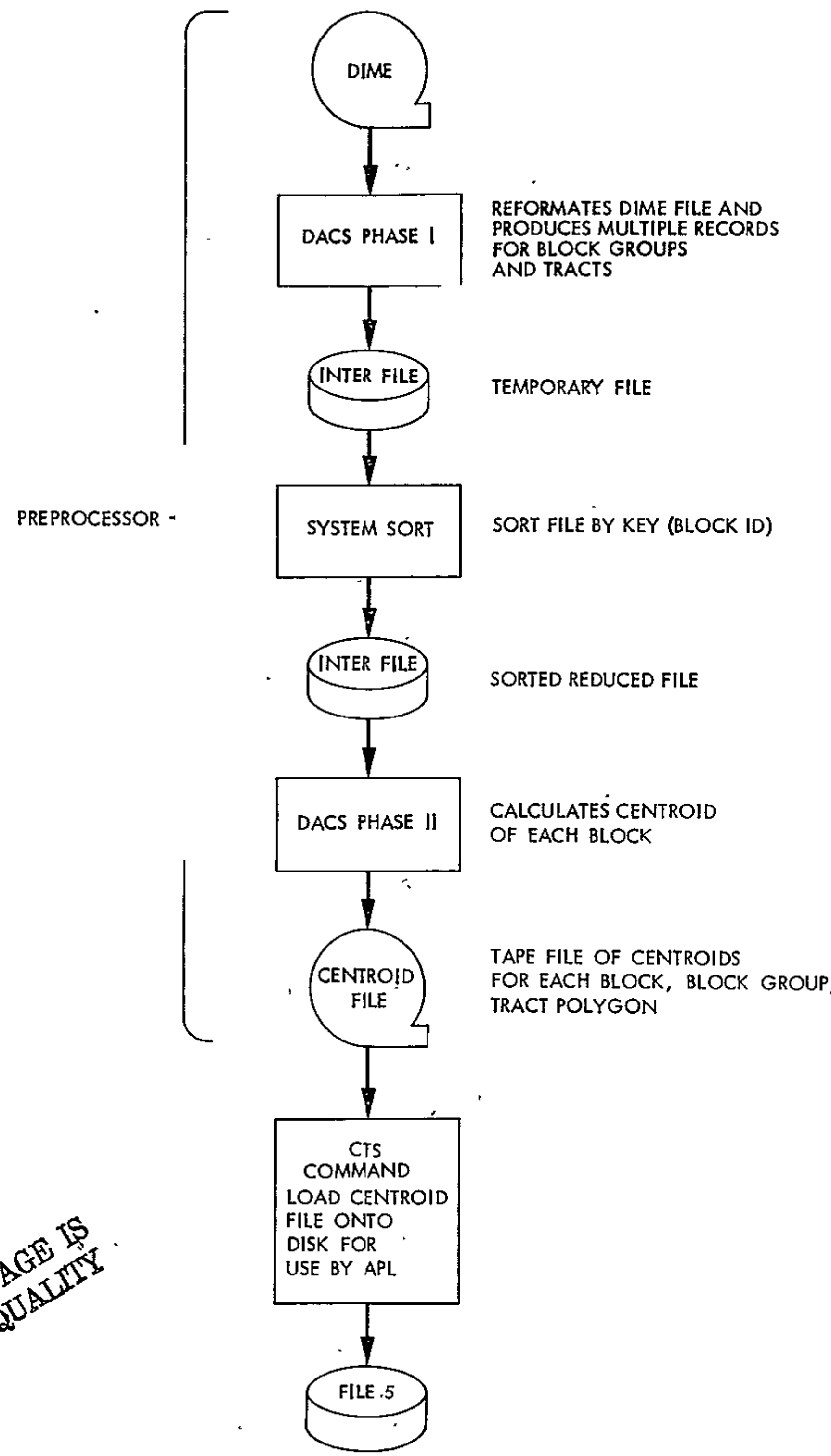
Subroutine CHAIN checks to see if the region's boundary segments form one or more closed loops. If they do not, a message to notify user of the condition is printed, and calculations are skipped. If they form more than one closed loop, another message is printed; however, in this case, calculation is allowed to proceed.

Data for a valid region is used in subroutine CALC, which computes the area and centroid coordinates. Subroutine POLYPT then determines whether or not the centroid lies within the region boundary. If it lies outside the boundary, an appropriate message is printed, and the centroid "x" and "y" coordinates are set equal to those of the nearest boundary segment node. Area and centroid are written onto the final output tape.

3.4 Picture Data File Processor (Note Fig. 3-4)

3.4.1 PDRP. PDRP is a FORTRAN Program to create RDIME file. It takes DIME file as input and reads following data:

1) Coding Limit Flag.



ORIGINAL PAGE IS OF POOR QUALITY

Fig. 3-3. Centroid Coordinate System

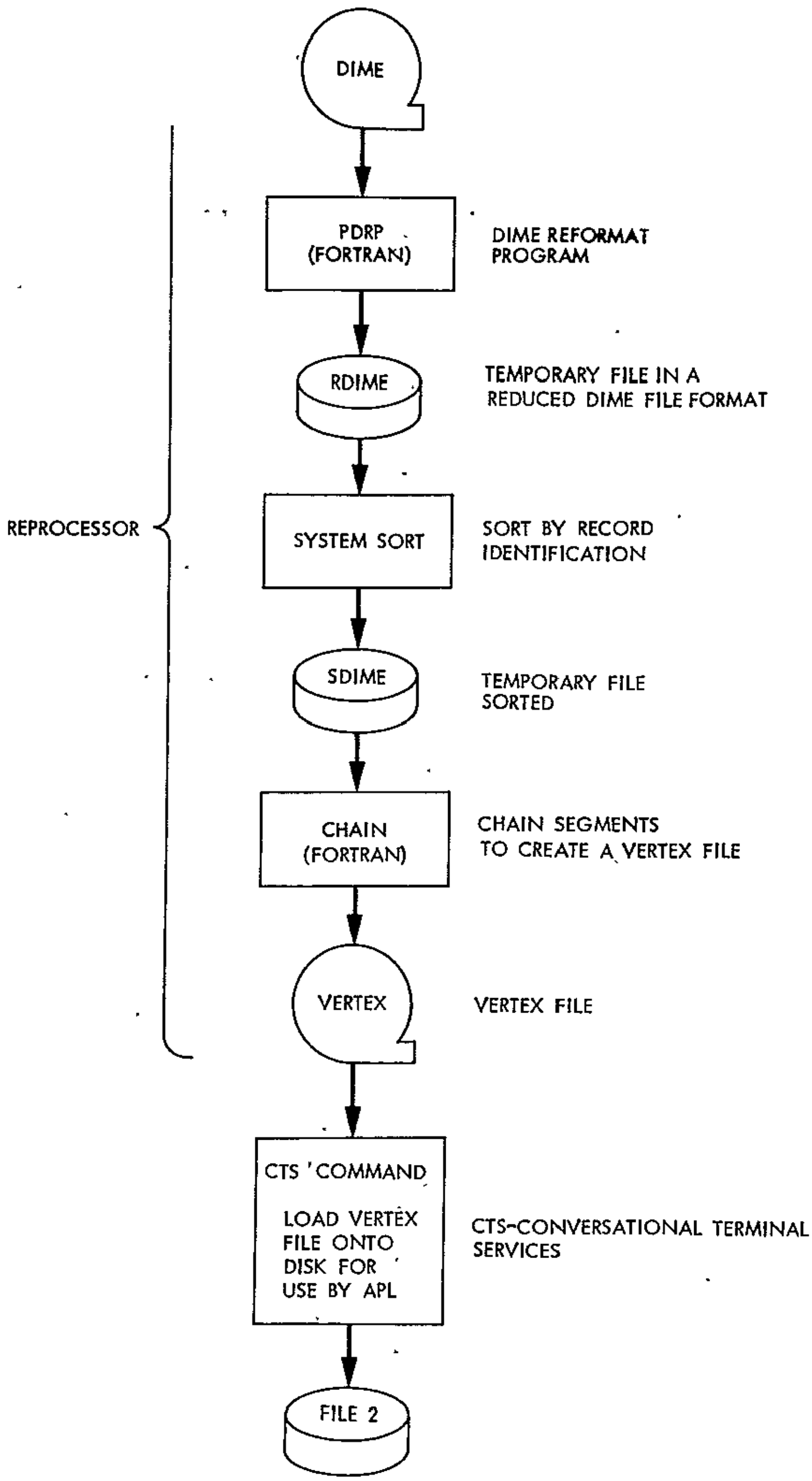


Fig. 3-4. Picture Data

2) Left/Right Tract Numbers and Suffices.

3) Left/Right Block Numbers and Suffices.

4) From/To xy Coordinates.

PDRP detects and counts erroneous records by testing Coding Limit Flag. PDRP assigns level numbers to each record according to the Left/Right Tract Numbers and Block Numbers as follows:

Level	Description
1	Block boundary
2	Group boundary
3	Tract boundary

The selected items are written to the temporary disk file, RDIME.

3.4.2 System Sort. The reformatted DIME File is sorted by the record identification. The sort produces another temporary field, SDIME (Sorted DIME file).

3.4.3 CHAIN. CHAIN takes SDIME as input and creates Vertex File of street segments. The purpose of CHAIN is to tie each segment one after another in order to facilitate the mapping function. Each segment is chained according to the level of boundary that the segment is representing.

3.5 APL File Creation and Update Program Description

3.5.1 CRTAB. CRTAB takes File-1 as input and creates the File 10, series files 11, 12, 13, Tabular Data Bases. The Block data of a group become a matrix of a Group data and stores as a record of an APL file, File 11. The row of matrix represents each block and the column represents data items. The column sum of data items of all blocks within a group becomes the group data, File 12, and the columns sum of data items of all groups within a Tract becomes Tract data, File 13.

3.5.2 CRIDX. CRIDX takes File-5, Centroid File, as an input and creates the File-30 series, Index File. The centroid file created in the preprocessor

step is divided into three files; index to the block, index to the block group, and index to tract.

3.5.3 CRPIC. CRPIC takes File-2 as input and creates the File-20 Series, Picture Data Base. All the level 1 data are grouped by Group level and stored sequentially in a record of an APL file. Each chain with levels of 2 and 3 are stored separately as a record of an APL file.

4.0 FILE DESCRIPTION

4.1 Third Count Summary Tape

Third Count Summary Tape is originally created by Bureau of Census and used by LUMIS-IV as an Input File. See APPENDIX-B for the Data Record Layout of Third Count Summary Tape.

4.2 USERS' DATA FILE

USER DATA FILES may be provided by user if necessary. The data in this file should be aggregated at the Block Level. The number of data field is not limited; however, each record must include Tract Number with Suffix and Block Number in the first data field of records.

Record No.	Shape		Contents
1 - N	N X 2	Row-1; Col-1:	Tract No. Suffix, Group No. and Block No. TTTTSSGBB.
		Row-2 through Row-N: Col-1:	X-coordinate Y-coordinate

4.3 File-1 (Temporary Tabular)

File-1 is a temporary APL file created as input to the APL function CRTAB. File-1 is basically similar to Third Count Summary Tape except that File-1 is an APL accessible file and contains only integer data. See APPENDIX-B for the Data Record Layout.

4.4 File-2 (Temporary Picture)

File-2 is a temporary APL file created as Input file to the APL function CRPIC. After File-20 is created by CRPIC, the contents of File-2 is dumped to the external storage File-4 as back-up data file and File-2 is deleted.

Record No.	Shape		Contents
1 - NT	N X 6	Col-1:	Tract number/Suffix and Block number/Suffix as TTTTSBBBS.
		Col-2:	Record number as RRRRRR.
		Col-3:	Level number as L.
		Col-4:	From State Plane Y-coordinate as YYYYYYY.
		Col-5:	From State Plane X-coordinate as XXXXXXX.
		Col-6:	To State Plane y-coordinate as YYYYYYY.
		Col-7:	To State Plane x-coordinate as XXXXXXX.

4.5 File-3 and File-4 (External Back-Up)

File-3 and File-4 are external tape files created by a system DUMP and contains the contents of File-1 and File-2.

4.6 File-10 Series (Tabular Files)

The 10 series of files are the Data Base File of LUMIS-IV. They contain data from the Third Count Census tapes and other sources depending on the individual user's requirements. The files are broken down by their levels of aggregation, block, block group, and tract.

File No.	Shape	Contents
11	NB X ND	Block Level Data Fields
12	NB X ND	Group Level Data Fields
13	NB X ND	Tract Level Data Fields

4.7 File-20 Series (Picture Files)

The 20 series of files contain the State Plane Coordinates obtained from the DIME file processed by CHAIN and CRPIC into four files.

File No.	Shape	Contents
21	NG x NV x 2	Block Dividing Segments in a Group (Lev=1)
22	NG x 10	Group Outline Index
23	NG x 10	Tract Outline Index
24	N x NV x 2	Tract Dividing Segments: (Lev= 2, 3)

4.8 File-30 Series (Index Files)

The 30 series of files contain the block, group and tract identification numbers of Bureau of the Census along with the centroids and pointer of their data locations of the 10 and 20 file series.

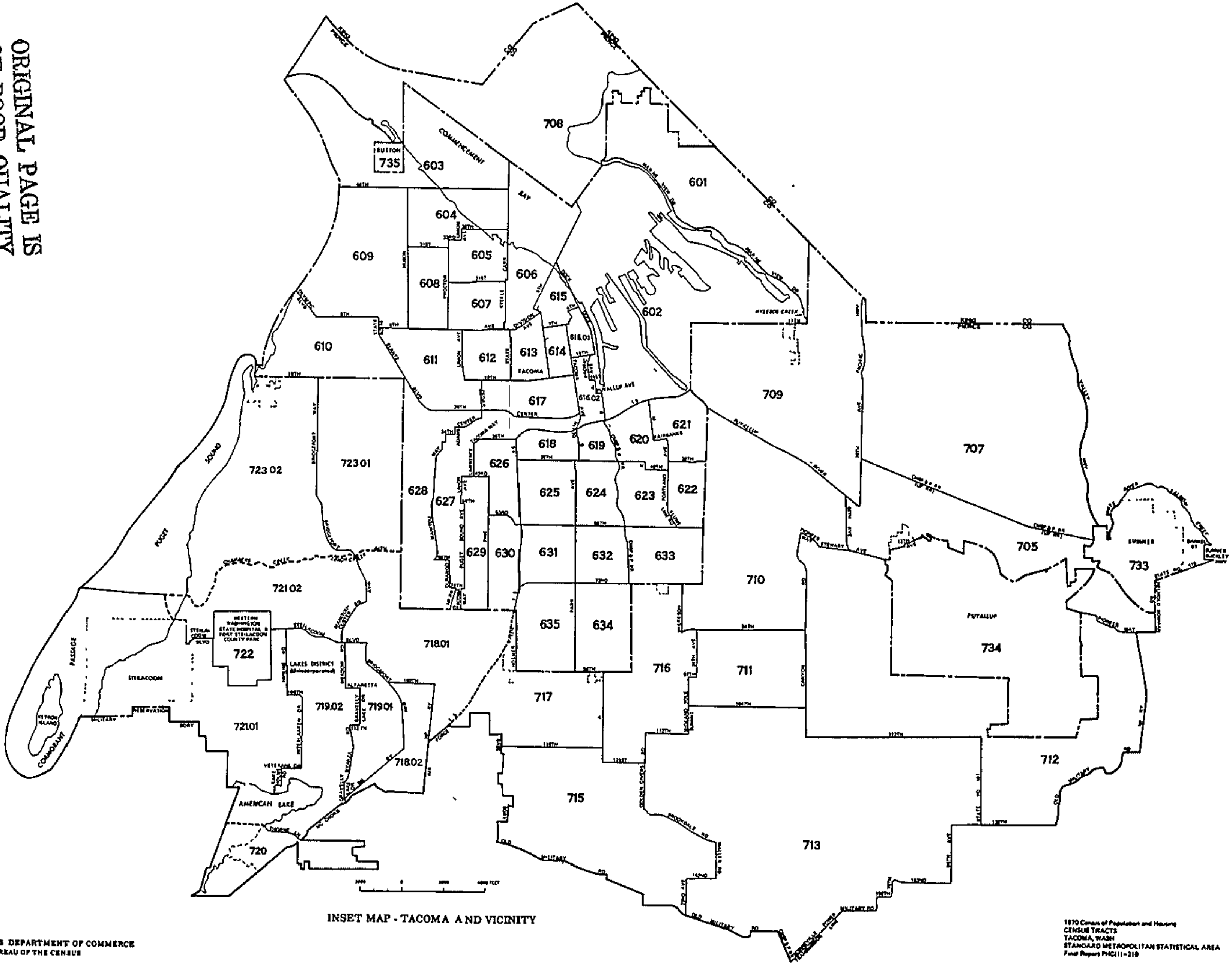
File No.	Shape	Contents	
31	NB x 3	Col-1:	Centroid Y
		Col-2:	Centroid X
		Col-3:	Block ID as TTTTSBBBS.
32	NB x 5	Col-1:	Centroid Y
		Col-2:	Centroid X
		Col-3:	Group ID as TTTTSG
		Col-4:	Block beginning pointer as NNNNN
		Col-5:	Number of block as NNNNN
33	NB x 5	Col-1:	Centroid Y
		Col-2:	Centroid X
		Col-3:	Tract ID is TTTTS
		Col-4:	Group beginning pointer as NNNNN
		Col-5:	Number of group as NNNNN

APPENDIX A

EXAMPLE OF A CENSUS TRACT REFERENCE MAP

ORIGINAL PAGE IS
OF POOR QUALITY

A-2



INSET MAP - TACOMA AND VICINITY

U. S. DEPARTMENT OF COMMERCE
BUREAU OF THE CENSUS

1970 Census of Population and Housing
CENSUS TRACTS
TACOMA, WASH.
STANDARD METROPOLITAN STATISTICAL AREA
Final Report PHC(1)-210

APPENDIX B

1970 THIRD-COUNT SUMMARY TAPE DATA RECORD LAYOUT

APPENDIX B

1970 Third-Count Summary Tape Data Record Layout

Table of Contents

APPENDIX B
Table of Contents

GENERAL FORMAT

The FORTRAN format statement of the data record is:

A1, A2 3X, 16X, A4, A2, 6X, A1, 32X, 12X, 3A2, 6A1, 6X, A3, 3X, A4, A2, A3, 2X, A1, 12I12, 233I6, 18X.

where:

rAw = Alphanumeric field
rIw = Numeric data field
wX = Padding (space) field
, = Field separator
r = Repeat count for field (a blank = 1 field)
w = Field length

Data No.	Data Description	Format	Field Position
1	Record Type	A1	1
2	1970 State	A2	2-3
-	Padding	3X	4-6
3	1960 State	A2	7-8
-	Padding	16X	9-24
4	Tract (Basic)	A4	25-28
5	Tract (Suffix)	A2	29-30
-	Padding	6X	31-36
6	Central Business District	A1	37
-	Padding	32X	38-69
7	1970 County of Tabulation	A3	70-72
-	Padding	12X	73-84
8-9	Sequencing Keys	6X	85-96
-	Padding	3A2, 6A1	97-102
10-13	Sequencing Keys	A3, 3X, A4, A2, A3	103-117
-	Padding	2X	118-119
14	$ Symbol	A1	120

Record Types are as follows:

0 = Summary is inside the UA
2 = Summary is in a Contract Block Area

These codes are used to control the publication of the summaries in this file.

Sequencing Keys contain various Geographic Codes as determined by the designated summary level (also see Page B-2).

Summary Categories and Sequencing Keys

	85-90-	91-96-	103-106-	109-114	115-120

URBANIZED AREAS (BLOCK PUBLISHING AREAS)

Tract	bbAAAAbb9999CCCbbbDDDDEEbbbbb$
Block	bbAAAAbb9999CCCbbbDDDDEEFFFbb$

CONTACT BLOCK PUBLISHING AREAS

Tract or Block Numbering Area	GGHHII 1/ CCCbbbDDDDEEbbbb$
Block	GGHHII 1/ CCCbbbDDDDEEFFFbb$

AAAA = Block Publishing Area (UA)
CCC = County
DDDD = Tract or, in untracted areas, Block Numbering Area (BNA)
EE = Tract Suffix (if applicable)
FFF = Block
GG = 1960 State
HH = Contract Block Area (CBA)
II = Sub-Contract Block ARea 2/

1/ The contents of charcters 91-96 are variable. This Summary Area key serves to identify and/or sequence summaries component to the various Block Publishing Area reports. Possible content are:

bbJJJJ = Designates a summary for a Tract of Block Component of a Contract Place

CCCKKK = Where CCC represents Contract County and KKK represents MCD or CCd.

999999 = Other Contract Area or Vicinity.

2/ These six digits identify each Contract Block Area

Data No.	Data Description		Format	Field Position
	AGGREGATE $ VALUE:			
15	Total owner occupied		I12	121-132
16	Negro owner occupied		I12	133-144
17	Vacant for sale only		I12	145-156
	AGGREGATE $ MONTHLY CONTRACT RENT:			
18	Total renter occupied		I12	157-168
19	Negro renter occupied		I12	169-180
20	Vacant for rent		I12	181-192
	AGGREGATE $ VALUE FOR UNITS WITH ALL PLUMBING FACILITIES:			
21	Total owner occupied		I12	193-204
22	Negro owner occupied		I12	205-216
23	Vacant for sale only		I12	217-228
	AGGREGATE $ MONTHLY CONTRACT RENT FOR UNITS WITH ALL PLUMBING FACILITIES:			
24	Total renter occupied		I12	229-240
25	Negro renter occupied		I12	241-252
26	Vacant for rent		I12	253-264
	POPULATION-RACE AND SEX:			
27	Total:	Male	I6	265-270
28		Female	I6	
29	Negro:	Male	I6	
30		Female	I6	
31	Other:	Male	I6	
32		Female	I6	295-300
	POPULATION-AGE AND SEX:			
33	Male:	Under 5 years	I6	
34		5	I6	
35		6	I6	
36		7-9	I6	
37		10-13	I6	325-330
38		14	I6	
39		15	I6	
40		16	I6	
41		17	I6	
42		18-19	I6	355-360
43		20	I6	
44		21	I6	
45		22-24	I6	
46		25-34	I6	

Data No.	Data Description	Format	Field Position
47	35-44	I6	385-390
48	45-54	I6	
49	55-59	I6	
50	60-61		
51	62-64	I6	
52	65-74	I6	415-420
53	75 and over	I6	421-426
54	Female: Under 5 years	I6	427-432
55	5	I6	
56	6	I6	
57	7-9	I6	445-450
58	10-13	I6	
59	14	I6	
60	15	I6	
61	16	I6	
62	17	I6	475-480
63	18-19	I6	
64	20	I6	
65	21	I6	
66	22-24	I6	
67	25-34	I6	505-510
68	35-44	I6	
69	45-54	I6	
70	55-59	I6	
71	60-61	I6	
72	62-64	I6	535-540
73	65-74	I6	
74	74 and over	I6	

POPULATION-14 YEARS OLD AND OVER BY MARITAL STATUS AND SEX:

Data No.	Data Description	Format	Field Position
75	Male: Now Married (excludes sep.)	I6	553-558
76	Widowed	I6	559-564
77	Divorced	I6	565-570
78	Separated	I6	
79	Never Married	I6	
80	Female: Now Married (excludes sep.)	I6	
81	Widowed	I6	
82	Divorced	I6	595-600
83	Separated	I6	601-606
84	Never Married	I6	607-612

RELATIONSHIP:

Data No.	Data Description	Format	Field Position
85	Head of Household	I6	613-618
86	Wife of head	I6	
87	Child of head	I6	625-630
88	Other relative of head	I6	
89	Nonrelative (includes roomer, boarder, or lodger) of head in household	I6	

Data No.	Data Description	Format	Field Position
90	Inmate of institution in group quarters	I6	
91	Other in group quarters	I6	643-648
	POPULATION UNDER 18 BY RELATIONSHIP AND TYPE OF FAMILY:		
92	Head or wife of head of household	I6	649-654
-	Own (never married) child of head:	I6	
93	In Husband-wife family	I6	
94	In other family with male head	I6	655-660
95	In family with female head	I6	
96	Other relative of head	I6	
97	Nonrelative (includes roomer, boarder, or lodger) of head in household	I6	685-690
98	In group quarters	I6	691-696
	TOTAL HOUSING UNITS:		
99	Total housing units	I6	697-702
	OCCUPANCY/VACANCY STATUS:		
100	Occupied	I6	703-708
101	Vacant year-round: For rent	I6	
102	For sale only	I6	
103	Other vacant	I6	721-726
	TENURE AND RACE OF HEAD:		
104	Owner Occupied: Total	I6	727-732
105	White	I6	
106	Negro	I6	
107	Renter Occupied: Total	I6	745-750
108	White	I6	
109	Negro	I6	757-762
	TYPE OF STRUCTURE:		
110	1-unit structure	I6	763-768
111	2-or-more-unit structures	I6	
112	Mobile homes or trailers (occupied only)	I6	775-780
	ROOMS IN UNIT:		
113	1 room in unit.	I6	781-786
114	2 rooms in unit	I6	
115	3 rooms in unit	I6	
116	4 rooms in unit	I6	
117	5 rooms in unit	I6	805-810
118	6 rooms in unit	I6	811-816

Data No.	Data Description	Format	Field Position
	AGGREGATE NUMBER OF ROOMS 1/ BY TENURE AND RACE OF HEAD:		
119	Total occupied and vacant year-round units	I6	817-822
120	Total occupied	I6	
121	Owner occupied		
122	Renter occupied	I6	835-840
123	Total Negro occupied	I6	
124	Negro owner occupied	I6	
125	Negro renter occupied	I6	853-858
	PERSONS IN UNIT:		
126	1 person in unit	I6	859-864
127	2 persons in unit	I6	865-870
128	3 persons in unit	I6	
129	4 persons in unit	I6	
130	5 persons in unit	I6	
131	6 persons in unit	I6	889-894
	AGGREGATE NUMBER OF PERSONS 1/ BY TENURE AND RACE OF HEAD:		
132	Total occupied	I6	895-900
133	Owner occupied	I6	
134	Renter occupied	I6	
135	Total Negro occupied	I6	
136	Negro owner occupied	I6	
137	Negro renter occupied	I6	925-930
	PERSONS PER ROOM, TENURE AND RACE OF HEAD:		
138	Total occupied: 1.00 or less persons per room	I6	931-936
139	1.01 - 1.50	I6	
140	1.51 or more	I6	
141	Owner occupied: 1.00 or less persons per room	I6	
142	1.01 - 1.50	I6	955-960
143	1.51 or more	I6	
144	Renter occupied: 1.00 or less persons per room	I6	
145	1.01 - 1.50	I6	
146	1.51 or more	I6	
147	Total Negro occupied: 1.00 or less persons per room	I6	985-990
148	1.01 - 1.50	I6	
149	1.51 or more	I6	

Data No.	Data Description	Format	Field Position
150	Negro owner occupied: 1.00 or less persons per room	I6	
151	1.01 - 1.50	I6	
152	1.51 or more	I6	1015-1020
153	Negro renter occupied: 1.00 or less persons per room	I6	
154	1.01 - 1.50	I6	
155	1.51 or more	I6	1033-1038
	NUMBER OF UNITS AT ADDRESS:		
156	2-4 units	I6	
157	5-9 units	I6	1045-1050
158	10 or more units	I6	1051-1056
	TOILET FACILITIES:		
159	Flush toilet for this household only	I6	1057-1062
160	Flush toilet but also used by another household	I6	
161	No flush toilet	I6	1075-1080
162	UNITS WITH A BASEMENT:	I6	1075-1080
163	UNITS LACKING COMPLETE KITCHEN FACILITIES FOR THEIR HOUSEHOLD ONLY:	I6	1081-1086
164	UNITS LACKING DIRECT ACCESS:	I6	1087-1092
	VALUE OF UNITS:		
165	Less than $ 5,000	I6	1093-1098
166	$ 5,000 - $ 9,999	I6	
167	$10,000 - $14,999	I6	1105-1110
168	$15,000 - $19,999	I6	
169	$20,000 - $24,999	I6	
170	$25,000 - $34,999	I6	
171	$35,000 - $49,999	I6	
172	$50,000 or more	I6	1135-1140
	COUNT OF UNITS:		
173	Total owner occupied	I6	1141-1146
174	Negro ower occupied	I6	
175	Vacant for sale only	I6	1153-1158
	MONTHLY CONTRACT RENT:		
176	With cash rent: Less than $40	I6	1159-1164
177	$ 40 - $ 59	I6	1165-1170
178	$ 60 - $ 79	I6	

Data No.	Data Description	Format	Field Position
179	$ 80 - $ 99	I6	
180	$100 - $119	I6	
181	$120 - $149	I6	
182	$150 - $199	I6	1195-1200
183	$200 or more	I6	
184	Without payment of cash rent	I6	1207-1212
	COST OF UNITS:		
185	Total renter occupied	I6	1213-1218
186	Negro renter occupied	I6	
187	Vacant for rent	I6	1225-1230
188	UNITS FOR RENT THAT HAVE BEEN VACANT LESS THAN 2 MONTHS	I6	1231-1236
189	UNITS FOR SALE ONLY THAT HAVE BEEN VACANT LESS THAN 6 MONTHS	I6	1237-1242
190	VACANT-YEAR ROUND UNITS THAT HAVE BEEN VACANT 6 MONTHS OR MORE	I6	1243-1248
	TYPE OF HOUSEHOLD:		
191	Husband-wife family	I6	1249-1254
192	Other family with male head	I6	1255-1260
193	Family with female head	I6	
194	Male primary individual	I6	
195	Female primary individual	I6	1273-1278
196	UNITS WITH ROOMERS, BOARDERS, OR LODGERS:	I6	1279-1284
	PLUMBING FACILITIES, TENURE AND RACE OF HEAD:		
	Total occupied and vacant year-round:		
197	With all plumbing facilities	I6	1285-1290
198	Lacking one or more plumbing fac.	I6	
	Total occupied:		
199	With all plumbing facilities	I6	
200	Lacking one or more plumbing fac.	I6	

Data No.	Data Description	Format	Field Position
	Owner occupied:		
201	With all plumbing facilities	I6	
202	Lacking one or more plumbing fac.	I6	1315-1320
	Renter occupied		
203	With all plumbing facilities	I6	
204	Lacking one or more plumbing fac.	I6	
	Total Negro occupied:		
205	With all plumbing facilities	I6	
206	Lacking one or more plumbing fac.	I6	
	Negro owner occupied		
207	With all plumbing facilities	I6	1345-1350
208	Lacking one or more plumbing fac.	I6	
	Negro renter occupied		
209	With all plumbing facilities	I6	
210	Lacking one or more plumbing fac.	I6	1363-1368
	FAMILIES BY PLUMBING FACILITIES:		
211	With all plumbing facilities	I6	1369-1374
212	Lacking one or more plumbing fac.	I6	1375-1380
	UNITS WITH 1.01 OR MORE PERSONS PER ROOM BY PLUMBING FACILITIES, TENURE AND RACE OF HEAD:		
	Total occupied:		
213	With all plumbing facilities	I6	1381-1386
214	Lacking one or more plumbing fac.	I6	
	Owner occupied:		
215	With all plumbing facilities	I6	
216	Lacking one or more plumbing fac.	I6	
	Renter occupied:		
217	With all plumbing facilities	I6	1405-1410
218	Lacking one or more plumbing fac.	I6	

Data No.	Data Description	Format	Field Position
	Total Negro occupied:		
219	With all plumbing facilities	I6	
220	Lacking one or more plumbing fac.	I6	
	Negro owner occupied:		
221	With all plumbing facilities	I6	
222	Lacking one or more plumbing fac.	I6	1435-1440
	Negro renter occupied:		
223	With all plumbing facilities	I6	
224	Lacking one or more plumbing fac.	I6	1447-1452
	VALUE FOR UNITS WITH ALL PLUMBING FACILITIES:		
	COUNT OF OWNER-OCCUPIED UNITS:		
225	Less than $ 5,000	I6	1453-1458
226	$ 5,000 - $ 9,999	I6	
227	$10,000 - $14,999	I6	1465-1470
228	$15,000 - $19,999	I6	
229	$20,000 - $24,999	I6	
230	$25,000 - $34,999	I6	
231	$35,000 - $49,999	I6	
232	$50,000 or more	I6	1495-1500
	COUNT OF UNITS:		
233	Total owner occupied	I6	1501-1506
234	Negro owner occupied	I6	
235	Vacant for sale only	I6	1513-1518
	MONTHLY CONTRACT RENT FOR UNITS WITH ALL PLUMBIN FACILITIES:		
236	With cash rent: Less than $40	I6	1519-1524
237	$ 40 - $ 59	I6	1525-1530
238	$ 60 - $ 79	I6	
239	$ 80 - $ 99	I6	
240	$100 - $119	I6	
241	$120 - $149	I6	
242	$150 - $199	I6	1555-1560
243	$200 or more	I6	
244	Without payment of cash rent	I6	1567-1572

Data No.	Data Description	Format	Field Position
	COUNT OF UNITS:		
245	Total renter occupied	I6	1573-1578
246	Negro renter occupied	I6	
247	Vacant for rent	I6	1585-1590
	POPULATION IN UNITS WITH 1. 01 OR MORE PERSONS PER ROOM BY TENURE:		
248	Total occupied	I6	1591-1596
249	Owner occupied	I6	
250	Renter occupied	I6	
251	Total Negro occupied	I6	
252	Negro owner occupied	I6	1615-1620
253	Negro renter occupied	I6	1621-1626
	POPULATION IN UNITS BY PLUMBING FACILITIES:		
254	With all plumbing facilities	I6	1629-1632
255	Lacking one or more plumbing fac.	I6	1633-1638
	POPULATION IN UNITS WITH 1. 01 OR MORE PERSONS PER ROOM BY PLUMBING FACILITIES:		
256	With all plumbing facilities	I6	1639-1644
257	Lacking one or more plumbing fac.	I6	1645-1650
258	NUMBER OF UNITS WITH CONTACT RENT ALLOCATED:	I6	1651-1656
259	NUMBER OF ALLOCATED OCCUPIED AND VACANT YEAR-ROUND HOUSING UNITS:	I6	1657-1662
	Padding		1663-1680

APPENDIX C

DIME FILE DATA RECORD LAYOUT

APPENDIX-C DIME FILE DATA RECORD LAYOUT

Data No.	Data Description	Data Length	Record Position	Data Type
1	Street Prefix Direction	2	1-2	A
2	Street or Non-Street Feature Name	20	3-22	AN
3	Street Type	4	23-26	A
4	Street Suffix	2	27-28	A
5	Non-Street Feature Code	1	29	AN
6	1970 Enumeration District Left (1970 Non-Mail Census Areas Only)	5	30-34	AN
7	Blank (Census Use Only)	6	35-40	B
8	1970 Enumeration District Right (1970 Non-Mail Census Areas Only)	5	41-45	AN
9	From Map (Basic Number)	3	46-48	N
10	From Map (Suffix)	2	49-50	A
11	To Map (Basic Number)	3	51-53	N
12	To Map (Suffix)	2	54-55	A
13	Coding Limit Flag	1	56	AN
14	Left Low Address	6	57-62	N
15	Left High Address	6	63-68	N
16	Right Low Address	6	69-74	N
17	Right High Address	6	75-80	N
18	File Code	4	81-84	N
19	Record Number	6	85-90	N
20	Check Digit	1	91	N
21	Census Tract Left (Basic)	4	92-95	N
22	Census Tract Left (Suffix)	2	96-97	N
23	Census Tract Right (Basic)	4	98-101	N
24	Census Tract Right (Suffix)	2	102-103	N

Data No.	Data Description	Data Length	Record Position	Data Type
25	ZIP Code Left	5	104-108	N
26	ZIP Code Right	5	109-113	N
27	SMSA	4	114-117	N
28	Street Code (1970 Mail Census Areas Only)	5	118-122	N
29	From Node	4	123-126	N
30	To Node	4	127-130	N
31	Place Code Left	4	131-134	N
32	Place Code Right	4	135-138	N
33	State Code Left	2	139-140	N
34	County Code Left	3	141-143	N
35	Minor Civil Division Code/Census County Division Code Left	3	144-146	N
36	1970 Congressional District Left	2	147-148	N
37	1970 Area Code Left	3	149-151	N
38	Block Left (Basic)	3	152-154	N
39	Block Left (Suffix)	2	155-156	N
40	1960-1970 Annexation Code Left (1970 Mail Census Areas Only)	1	157	AN
41	State Code Right	2	158-159	N
42	County Code Right	3	160-162	N
43	Minor Civil Division	3	163-165	N
44	1970 Congressional District Right	2	166-167	N
45	1970 Area Code Right	3	168-170	N
46	Block Right (Basic)	3	171-173	N
47	Block Right (Suffix)	2	174-175	N
48	1960-1970 Annexation Code Right (1970 Mail Census Areas Only)	1	176	AN

Data No.	Data Description	Data Length	Record Position	Data Type
49	From State Plane Code	2	177-178	N
50	To State Plane Code	2	179-180	N
51	From Map Set Mile (X Coordinate)	6.3	181-186	N
52	From Map Set Mile (Y Coordinate)	6.3	187-192	N
53	To Map Set Mile (X Coordinate)	6.3	193-198	N
54	To Map Set Mile (Y Coordinate)	6.3	199-204	N
55	From Latitude (Y Coordinate)	6.4	205-210	N
56	From Longitude (X Coordinate)	7.4	211-217	N
57	To Latitude (Y Coordinate)	6.4	218-223	N
58	To Longitude (X Coordinate)	7.4	224-230	N
59	From State Plane (Y Coordinate)	7	231-237	N
60	From State Plane (X Coordinate)	7	238-244	N
61	To State Plane (Y Coordinate)	7	245-251	N
62	To State Plane (X Coordinate	7	252-258	N
63	Blank (Census Use Only)	42	259-300	B

APPENDIX D

LUMIS INTERACTIVE GRAPHICS APL PROGRAM LISTINGS

```
∇AREA[⎕]∇
    ∇ APEA;A;I;J;T
[1]   0⍴⎕DL 1
[2]   'NEW AREA?
[3]   →0⍴ANS'NQY'
[4]   INDX←0 3⍴⍳0
[5]   CN←PB←PD←PP←⍳0
[6]   IDENT←RN←0 40⍴⍳0
[7]   ALL←I←1
[8]   NT←⍴TIF←(TIX←FREAD 33 1)[;3]÷10000
[9] L5:'LEVEL? '
[10]  A←1↑⍞
[11]  →(0=A∊'BGTQ')/L6
[12]  →(A='B','G','T','Q')/L10,L12,L30,0
[13] L6:'LEVELS ARE B,G OR T'
[14]  →L5
[15] L10:→L15,LEV←1
[16] L12:LEV←2
[17] L15:'TPACT AND GROUP NUMBERS?
[18] L20:J←⍕I
[19]  ⍞←J,': '
[20]  →((1↑3↓T←⍞)='Q',' ')/L70,0
[21]  →(0=∨/T∊NUMA)/L29
[22]  IDT←DECODE(1 1)⍴1↑T←⍎T
[23]  IDG←(1,⍴1↓T)⍴1↓T
[24]  →L35
[25] L29:'WRONG NUMBER'
[26]  →(LEV≠3)/L20
[27] L30:LEV←3
[28]  'TRACT NUMBER(S)? '
[29]  →((1↑T←⍞)='Q',' ')/L70,0
[30]  →(0=∧/T∊NUMA)/L29
[31]  IDT←DECODE(1,⍴⍎T)⍴⍎T
[32] L35:→(∧/L←NT≥PT←TIF⍳IDT)/L40
[33]  'TRACT ';((~L)/IDT)÷10;' ARE NOT IN FILE'
[34]  'TRY AGAIN'
[35]  →(LEV=1,2,3)/L20,L20,L30
[36] L40:→(LEV<3)/L50
[37]  INDX←INDX,[1]((⍴PT),3)⍴TIX[PT;⍳3]
[38]  CN←CN,1
[39]  RN←(1,⍴PT)⍴PT
[40]  J←1
[41]  PP←⍳0
[42]  Z←,RN
[43] L42:PP←PP,(FREAD 32,Z[J])[;4]
[44]  →((⍴Z)≥J←J+1)/L42
[45]  →0
[46] L50:NG←⍴A←100×10|⌊(GIX←FREAD 32,PT)[;3]÷1000
[47]  →(1≠1 1↑IDG)/L51
[48]  IDG←A
[49] L51:→(∧/NG≥PG←,A⍳IDG)/L55
[50]  'TRACT ';IDT÷10;' CONTAINS ';A;' GROUPS'
[51]  →L20
[52] L55:PP←PP,GIX[PG;4]
[53]  IDEN←⍕,(IDT÷10),IDG
[54]  IDENT←IDENT,[1](1 40)⍴IDEN,(40-(⍴IDEN))⍴' '
[55]  →(LEV=1)/L59
[56]  CN←CN,PT
[57]  PD←⍳0
[58]  PD←PD,PG
[59]  RN←RN,[1](1 40)⍴PD,(40-⍴PD)⍴0
[60]  INDX←INDX,[1]((⍴PG),3)⍴GIX[PG;⍳3]
[61]  →L65
[62] L59:J←1
[63]  CN←⍳0
[64]  CN←CN,GIX[PG;4]
[65]  PD←PD,GIX[PG;5]
[66] L60:INDX←INDX,[1](FREAD 31,CN[J])[;⍳3]
[67]  →((⍴CN)≥J←J+1)/L60
[68] L65:I←I+1
[69]  →L20
[70] L70:→0,FLGQ←1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
∇CENTROID[⎕]∇
    ∇ CENTROID;B;I;J;NB;NC;NT;T;Y;YY;Z
[1]   PREP
[2]   FLGQ←0
[3]  L1:AREA
[4]   →(FLGQ=1)/L40
[5]   99⌶27 12
[6]   ⎕DL 3
[7]   CHAIN
[8]   MIN VTX
[9]   GRAPH
[10]  NB←1↑⍴IDX
[11]  T←(NB,2)⍴MIN+GIN+FF
[12]  T[;1]←T[;1]-(0,⍳NB-1)×25÷FF
[13]  →(LEV=2 3)/L2,L3
[14]  99⌶(T←⌊0.5+T)NODE⍕B←(1000⊥0 2↓IDX)÷10
[15]  →L4
[16] L2:99⌶(T←⌊0.5+T[PG;])NODE⍕B←100×⌊(1000⊥0 2↓0 ¯2↓IDX[PG;])÷1000
[17]  NB←1↑⍴T
[18]  →L4
[19] L3:B←(0 2↓0 ¯2↓IDX[PT;])÷1000000
[20]  99⌶(T←⌊0.5+T[PT;])NODE⍕B←1 1⍴B←DECODE B
[21]  NB←1↑⍴T
[22] L4:J←1
[23] L5:→(∧/10>YX←GIN)/L40
[24]  →(GC='Q')/L40
[25]  Y←1 2⍴IDX[RN[J];⍳2]←⌊0.5+MIN+YX÷FF
[26]  99⌶Y NODE(1,⍴Z)⍴Z←⍕B[J;1]
[27]  99⌶(1 2⍴T[J;])NODE 1 5⍴'X'
[28]  →(NB≥J←J+1)/L5
[29] L40:99⌶29 56 107 32 64 31
[30]  ⍞←'SAVE? '
[31]  →L43×ANS'NY'
[32]  IDX FREPLACE(30+LEV),CN
[33] L43:'MORE? '
[34]  →L1×ANS'YN'
[35] L45:'END OF CENTROID'
[36]  FUNTIE FNUMS
```

```
∇COST[⎕]∇
    ∇ R←COST
[1]   (2 4⍴'CPU:CON:'),'8SZZ1.ZZ'⍕2 1⍴1 1↓⎕AI[2 3]÷2500 3600000
```

```
∇DECODE[⎕]∇
    ∇ R←DECODE X
[1]   R←10000 10 10⊤⌊100×,|X
[2]   R←10000 10⊥(¯2 0↓R),[1]⌈/(¯2,1↓⍴R)↑R
```

```
∇DISPLAY[⎕]∇
    ∇ DISPLAY
[1]   INIT
[2]  L5:OUTLINE
[3]   →(1=FLGQ)/L20
[4]   LOOK
[5]   →(1=FLGQ)/L20
[6]   →(TBL=0)/L15
[7]   TABLE
[8]   →L5
[9]  L15:MAP
[10]  →L5
[11] L20:0⍴⎕DL 1
[12]  FUNTIE FNUMS
[13]  'END OF PROCESS'
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
∇EDIT[⎕]∇
    ∇ EDIT;P
[1]    P←1 1⍴'·'
[2]    PREP
[3]   L5:FLGQ←0
[4]    FLGS←0
[5]    A̲REA
[6]    →(FLGQ=1)/L65
[7]    UTX←FREAD 21,PP
[8]   L13:ID←1 2↑UTX
[9]    UTX←1 0↓UTX
[10]   U←UTX
[11]  L15:M̲IN
[12]   99⌶27 12
[13]   DDL 2
[14]  L17:99⌶PIC←B̲LK U
[15]   99⌶29 56 107 32 64 31
[16]   'OK? '
[17]   →L60×A̲NS'YN'
[18]   EDITNODE
[19]   →(FLGS=0)/2+⍳26
[20]   U←(((1↑⍴U)-1↑⍴UTX),2)↑U
[21]   99⌶29 56 107 32 64 31
[22]   5⍴' '
[23]  L20:'WHAT? '
[24]   →((1↑⍞)='A','D','Q','R')/L30,L40,L65,L50
[25]   'TYPE A,D,Q OR R AND RETURN'
[26]   →L20
[27]  L30:FLG←0
[28]  L32:XY←⌊0.5+MIN+GIN+FF
[29]   →(GC='Q')/L58
[30]   99⌶(1 2⍴XY)N̲ODE P
[31]   'OK TO ADD? '
[32]   →L20×A̲NS'NY'
[33]   →(FLG≠0)/L35
[34]   UTX←UTX,[1]XY×¯1 1
[35]   FLG←1
[36]   →L32
[37]  L35:UTX←UTX,[1]XY
[38]   →L32
[39]  L40:'NODE NUMBER? '
[40]   →((1↑J←⍞)='Q')/L58
[41]   'OK TO DELETE NODE ';J;'?'
[42]   →L40×A̲NS'NY'
[43]   L←∨/L←(|UTX)∊|UT[⊥J;]
[44]   M←⍳1↑⍴L
[45]   UTX←UTX[L/M;]
[46]   →L40
[47]  L50:'NODE NUMBER? '
[48]   →((1↑J←⍞)='Q')/L58
[49]   XY←⌊0.5+MIN+GIN+FF
[50]   99⌶(1 2⍴XY)N̲ODE P
[51]   'OK TO REPLACE NODE ';J;'?'
[52]   →L50×A̲NS'NY'
[53]   L←∧/L←(|UTX)∊|UT[⊥J;]
[54]   LM←L/M←⍳1↑⍴L
[55]   K←1
[56]  L52:→(UTX[LM[K];1]<0)/L55
[57]   UTX[LM[K];]←XY
[58]   →((⍴LM)≥K←K+1)/L52
[59]   →L50
[60]  L55:UTX[LM[K];]←XY×¯1 1
[61]   →((⍴LM)≥K←K+1)/L52
[62]   →L50
[63]  L58:→(FLGS=1)/L
[64]   U←UTX
[65]   →L15
[66]  L59:U←U,[1]UTX
[67]   →L15
[68]  L60:'SAVE? '
[69]   →L61×A̲NS'NY'
[70]   →(FLGS=1)/L63
[71]   UTX←ID,[1]UTX
[72]   UTX FREPLACE 21,PP
[73]  L61:→(FLGS=1)/L62
[74]   FLGS←1
[75]   U←UTX
[76]   T←⍳0
[77]   T←T,FREAD 22,PP
[78]   T←T,FREAD 23,PP
[79]   I←0
[80]  L62:→((⍴T)<I←I+1)/L64
[81]   UTX←FREAD 24,T[I]
[82]   ID←1 2↑UTX
[83]   UTX←1 0↓UTX
[84]   U←U,[1]UTX
[85]   →L15
[86]  L63:UTX←ID,[1]UTX
[87]   UTX FREPLACE 24,T[I]
```

```
[88]  →L62
[89] L64:'MORE? '
[90]  →L5×⍙NS'YN'
[91]  FUNTIE FNUMS
[92] L65:'END OF EDIT'
    ∇
```

```
∇EDITNODE[⎕]∇
     ∇ EDITNODE
[1]    K←0
[2]    VT←0 2⍴⍳0
[3]   L10:→((1↑⍴VTX)<K←K+1)/L15
[4]    →(∧/L←(|VTX[K;])∊|VT)/L10
[5]    VT←VT,[1](1 2⍴VTX[K;])
[6]    →L10
[7]   L15:99⌶(|VT)⍙ODE⊤(Z,1)⍴⍳Z←1↑⍴VT
    ∇
```

```
∇INIT[⎕]∇
     ∇ INIT
[1]    →(10=(⍴FNUMS))/L1
[2]    FUNTIE FNUMS
[3]    TIEFILE
[4]   L1:FLGQ←TERM←TBL←EXPR←0
[5]    'GRAPHIC TERMINAL? '
[6]    →L2×⍙NS'YN'
[7]    TERM←1
[8]   L2:'MAP? '
[9]    →L3×⍙NS'YN'
[10]   TBL←1
[11] L3:'PRESS RETURN KEY TO CONTINUE'
[12]   ⍞
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
∇DATA[⎕]∇
    ∇ DATA;D;EX;EXP;I;JX;N;NM;NN;A;DE;T
[1]    INF←⍳0
[2]    LL←(1↑⍴INDX)⍴0
[3]   L1:'DATA? '
[4]    EX←EXP←,⎕
[5]    →((1↑EXP)='I',' ','Q')/L40,L50,L60
[6]    EXPR←EX
[7]   L2:NM←⍳0
[8]    →((⍴EXP)<I←EXP⍳'D')/L8
[9]   L3:NN←⍳0
[10]   EXP[I]←' '
[11]  L4:→((⍴EXP)<I←I+1)/L6
[12]   →(EXP[I]∊NUM)/L5
[13]   →(EXP[I]∊QP)/L6
[14]   →L8
[15]  L5:NN←NN,EXP[I]
[16]   →L4
[17]  L6:→(0=∧/(NN←⍎NN)∊⍳55)/L8
[18]   NM←NM,NN
[19]   →((⍴EXP)<I←EXP⍳'D')/L10
[20]   →L3
[21]  L8:'DATA IS D1 THRU D55'
[22]   →L1
[23]  L10:I←0
[24]   DAT←(0,¯1↑⍴NM)⍴⍳0
[25]  L14:→(LEV=1)/L16
[26]  L15:→((⍴CN)<I←I+1)/VA
[27]   PD←(0≠RN[I;])/RN[I;]
[28]   T←(FREAD(LEV+10),CN[I])[PD;NM]
[29]   →L17
[30]  L16:→((⍴PP)<I←I+1)/VA
[31]   T←(FREAD 11,PP[I])[⍳PD[I];NM]
[32]  L17:DAT←DAT,[1]((1↑⍴T),¯1↑⍴NM)⍴T
[33]   →(LEV=1 2)/L16,L15
[34]  VA:JX←0
[35]  L20:KX←JX+1
[36]   DN←'D',⍕NM[KX]
[37]   ⍎DN,'←(0,JX)↓((1↑⍴DAT),KX)↑DAT'
[38]   →((1↑⍴NM)>JX←JX+1)/L20
[39]   ⍎'LL←LL∨0>,',,DN
[40]   E←⍳0
[41]  L24:→((⍴EXP)<I←EXP⍳'÷')/L30
[42]   EXP[I]←' '
[43]   E←E,EX[⍳I-1]
[44]   E←E,' DIV '
[45]   E←E,I↓EX
[46]   EX←E
[47]   →L24
[48]  L30:DD←⍎EX
[49]   DD[LL/⍳⍴LL;]←9
[50]  L25:MM←(⌈/(~LL)/DD),⌊/(~LL)/DD
[51]   DEC←0
[52]   →(2=1↓ITYPE'DD')/INT
[53]   ⍞←'HOW MANY DECIMAL PLACES? '
[54]   →(2+⎕I26)+0×DEC←¯1↑ANS NUM
[55]   DEC←9
[56]  INT:DD←(((MM[2]<0)+DEC+(DEC≠0)+1+⌊10⍟⌊0.5+1⌈⌈/|MM),DEC)⍕DD
[57]   DD←(+/∧\DD=' ')⌽DD
[58]   DD[LL/⍳⍴LL;]←((+/LL),1↓⍴DD)⍴(1↓⍴DD)↑'CENSORED'
[59]   DD[INF;]←((⍴INF),1↓⍴DD)⍴(1↓⍴DD)↑'INFINITY'
[60]   A←(⌊0.5+MM×DE)÷DE←10*DEC
[61]   'MAXIMUM IS ',⍕A[1]
[62]   'MINIMUM IS ',⍕A[2]
[63]   'TITLE?'
[64]   TIT←,⍞
[65]   →0
[66]  L40:TIT←'CENSUS BUREAU ID'
[67]   DD←⍕DD←ID
[68]   →0
[69]  L50:→(EXPR=0)/L65
[70]   EX←EXP←EXPR
[71]   →L2
[72]  L60:→0,FLGQ←1
[73]  L65:'DATA NOT DEFINED'
[74]   →L2
    ∇
```

```
∇MAP[⎕]∇
    ∇ MAP;M;F
[1]    FF←÷ 10000000000
[2]    CHAIN
[3]    SCALE UTX
[4]    'PRESS RETURN KEY TO CONTINUE'
[5]    ⍞
[6]    99⌶27 12
[7]    0⍴⎕DL 1
[8]    GRAPH
[9]    99⌶INDX[;⍳2]⍙NODE DD
[10]   99⌶29 56 107 32 64 31
[11]   →(LEV<3)/L10
[12]   'TRACT: ',⍕IDT÷10
[13]   →L15
[14]  L10:'TRACT  GROUPS'
[15]   IDENT
[16]  L15:' '
[17]   TIT
[18]   M←4 2⍴P←GIN
[19]   99⌶⍙LK⌊0.5+((⍴M)⍴MIN)+(M+4 2⍴0 0 ¯10 0 ¯10 200 0 200)÷FF
[20]   M←⍕(⌊0.5+2000÷FF)÷10
[21]   99⌶(1 2⍴⌊0.5+MIN+(P+¯30 10)÷FF)⍙NODE(1,⍴M)⍴M
[22]   ' FEET'
[23]   ⍞
    ∇
```

```
∇PREP[⎕]∇
    ∇ PREP
[1]    →(7=1↑⍴FNUMS)/L5
[2]    TIE30
[3]    TIE40
[4]   L5:NT←⍴TIF←(TIX←FREAD 33 1)[;3]÷10000
    ∇
      ∇RESET[⎕]∇
    ∇ RESET
[1]    ⎕EX ⍙LPHA ⎕NL 2
    ∇
      ∇TABLE[⎕]∇
    ∇ TABLE;ID;F
[1]    →(TERM>0)/L5
[2]    'PRESS RETURN KEY TO CONTINUE'
[3]    ⍞
[4]    99⌶27 12
[5]    0⍴⎕DL 5
[6]    0⍴⎕DL 2
[7]   L5:ID←⍉100000 10 1000⊤IUA[;3]
[8]    ID[;1]←ID[;1]÷10
[9]    ID[;2]←ID[;2]×100
[10]   →((∧/F,ID[;2]=0),F←∧/ID[;3]=0)/TKS,GPS
[11]   ID[;3]←(ID[;3]÷10)+(ID[;2]×10)÷10
[12]   ' TRACT GROUP   BLOCK ',TIT
[13]   ('BF6.1,X3,BI3,X3,BF5.1,X3'⍺(ID[;1];ID[;2];ID[;3])),DD
[14]   →0
[15]  GPS:'TRACT GROUP ',TIT
[16]   ('BF6.1,X3,BI3,X3'⍺(ID[;1];ID[;2])),DD
[17]   →0
[18]  TKS:' TRACT '
[19]   ('BF6.1,X3'⍺(ID[;1])),DD
[20]   ⍞
    ∇
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
∇ANS[⎕]∇
     ∇ R←ANS C
[1]    →((⁻1 0+⍴C)≥R←C⍳1↑(R≠' ')/R←⍞)/0,E
[2]    ⍞←'PLEASE TYPE ',(,⍉(((1,⁻2+⍴C)⍴C),[1]',',')),' ',C[⁻1+⍴C],' OR ',(⁻1↑
C),': '
[3]    →1
[4]   E:R←⍳0
     ∇
      ∇AREA[⎕]∇
     ∇ AREA
[1]    PP←⍳0
[2]   L5:LEV←3
[3]    'TRACT NUMBER? '
[4]    →((1↑TN←,⍞)=' ','Q')/L25,L45
[5]    TN←DECODE⍎TN
[6]    →(∧/L←NT≥PT←TIF⍳TN)/L10
[7]    'TRACT ';((~L)/TN)÷10;' ARE NOT AVAILABLE'
[8]    →L5
[9]   L10:RN←⍳0
[10]   CN←1
[11]   RN←RN,PT
[12]   IDX←TIX
[13]   NG←⍴ALL←100×10|⌊(GIX←FREAD 32,PT)[;3]÷1000
[14]   →(2>1↑⍴TN)/L25
[15]   PP←PP,GIX[;4]
[16]   →0
[17]  L25:'GROUP NUMBER? '
[18]   →((1↑⍴GN←,⍞)=' .','Q')/L35,L45
[19]   →(∧/NG≥PG←,ALL⍳GN←⍎GN)/L30
[20]   'TRACT ';TN÷10;' CONTAINS ';ALL;' GROUPS'
[21]   →L25
[22]  L30:LEV←2
[23]   CN←PT
[24]   RN←⍳0
```

```
∇TIEFILE[⎕]∇
     ∇ TIEFILE
[1]    TIE10
[2]    TIE20
[3]    TIE30
     ∇
      ∇TIE10[⎕]∇
     ∇ TIE10
[1]    'TAB1'FTIE 11
[2]    'TAB2'FTIE 12
[3]    'TAB3'FTIE 13
     ∇
      ∇TIE20[⎕]∇
     ∇ TIE20
[1]    'PIC1'FTIE 21
[2]    'PIC2'FTIE 22
[3]    'PIC3'FTIE 23
[4]    'PIC4'FTIE 24
     ∇
      ∇TIE30[⎕]∇
     ∇ TIE30
[1]    'IDX1'FTIE 31
[2]    'IDX2'FTIE 32
[3]    'IDX3'FTIE 33
     ∇
      ∇TIE40[⎕]∇
     ∇ TIE40
[1]    'TEST1'FTIE 41
[2]    'TEST2'FTIE 42
[3]    'TEST3'FTIE 43
[4]    'TEST4'FTIE 44
     ∇
```

```
∇AREA[⎕25]∇
[25]   RN←RN,PG
[26]   IDX←GIX
[27]   PP←PP,GIX[PG;4]
[28]   →(2>1↑⍴GN)/L35
[29]   →0
[30] L35:'BLOCK LEVEL? '
[31]   →0×ANS'NY'
[32]   LEV←1
[33]   CN←PP
[34] L40:IDX←FREAD 31,PP
[35]   RN←⍳1↑⍴IDX
[36]   →0
[37] L45:FLGO←1
     ∇

      ∇BLK[⎕]∇
    ∇ R←BLK D;L;A
[1]    A←(0>D[;1])/⍳1↑⍴D
[2]    R←((2×⍴D)⍴32 96 32 64)+,⍉32 32⊤⌊FF×(|D)-(⍴D←,D)⍴MIN
[3]    R←(~(⍳(⍴R)+⍴A)∊L←A+⍳⍴A←4×0,¯1+A)\R
[4]    R[L]←29
    ∇


∇CHAIN[⎕]∇
    ∇ CHAIN;I;IX;T;N
[1]    VTX←0 2⍴⍳0
[2]    I←1
[3]    IX←⍳0
[4]    →(LEV=2 3)/L15,L25
[5]  L10:VTX←VTX,[1](1 0)↓FREAD 21,PP[I]
[6]    →((⍴PP)≥I←I+1)/L10
[7]  L15:I←1
[8]  L20:T←FREAD 22,PP[I]
[9]    IX←IX,(~L←T∊IX)/T
[10]   →((⍴PP)≥I←I+1)/L20
[11] L25:I←1
[12] L30:T←FREAD 23,PP[I]
[13]   IX←IX,(~L←T∊IX)/T
[14]   →((⍴PP)≥I←I+1)/L30
[15]   I←1
[16] L40:VTX←VTX,[1](1 0)↓FREAD 24,IX[I]
[17]   →((⍴IX)≥I←I+1)/L40
    ∇

      ∇DIV[⎕]∇
    ∇ R←A DIV B;T
[1]    B[INF←(0=,B)/⍳1↑⍴B;]←9
[2]    R←A÷B
    ∇

      ∇GIN[⎕]∇
    ∇ R←GIN
[1]    99⌶27 26
[2]    R←⌽(2 2⍴¯1+HXY⍳1↓GC←5↑⍞)+.×32 1
[3]    GC←GC[1]
    ∇


∇GRAPH[⎕]∇
    ∇ GRAPH;J
[1]    PIC←⍳0
[2]    PIC←PIC,BLK VTX
[3]    99⌶PIC
    ∇
      ∇ID[⎕]∇
    ∇ R←ID;LL
[1]    →(~∨/LL←0=,R←⌊0.1×1000⊥0 2↓INDX)/0
[2]    R[LL;1]←IVA[LL←LL/⍳⍴LL;3]÷100000
    ∇
      ∇MIN[⎕]∇
    ∇ MIN
[1]    MAX←⌈⌿VTX
[2]    MIN←⌊⌿VTX
[3]    FF←⌊/700 1023÷MAX-MIN
    ∇


      ∇NODE[⎕]∇
    ∇ R←YX NODE A;N
[1]    YX←0⌈⌊FF×YX-(⍴YX)⍴MIN
[2]    R←,29,((((1↑⍴YX),4)⍴(((1↑⍴YX),2 2)⍴32 96 32 64)+3 1 2⍉32 32⊤YX),31),
C⍳A
    ∇
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
∇SCALE[⎕]∇
    ∇ SCALE A;T
[1]    MAX←⌈/|A
[2]    MIN←⌊/|A
[3]    TT←(0.1×⌊0.5+129.87+FF←⌊/700 1023÷MAX-MIN)
[4]    CR,'NEW SCALE IS ';TT;' FEET/INCH'
[5]    'NEW LOWER LEFT CORNER IS:'
[6]    MIN[2];' FEET EAST BY'
[7]    MIN[1];' FEET NORTH.'
    ∇
       ∇TYPE[⎕]∇
    ∇ CODE←TYPE ARG
[1]    CODE←(⎕NC ARG),0
[2]    →(((1↑CODE)=2),⌈((1↑CODE)=3))/VAR,F'
[3]    →0
[4]   VAR:CODE[2]←(1 2 3 0)[1⍴3I⍎ARG]
[5]    →0
[6]   FN:CODE[2]←0≠+/⍴⎕CR ARG
    ∇


∇CONVC[⎕]∇
    ∇ R←CONVC C;I;L
[1]    I←⍳⍴L←C∊'-'
[2]    C[L/I]←'0'
[3]    R←⍎C
    ∇
       ∇CONVP[⎕]∇
    ∇ CONVP;I;L
[1]    I←⍳⍴L←A∊'-'
[2]    A[L/I]←'¯'
[3]    A←⍎A
    ∇
       ∇CONVT[⎕]∇
    ∇ CONVT;I;L
[1]    I←⍳⍴L←T∊'-'
[2]    T[L/I]←'¯'
[3]    T←⍎T
[4]    T←T[⍳RL]
    ∇


∇CRCENT[⎕]∇
    ∇ CRCENT;TAPE1;A;R
[1]    DEFILE
[2]    CENT←0 3⍴⍳0
[3]    A←TAPE1
[4]   L5:A←TAPE1
[5]    R←CONVC A
[6]    →(9999<R[1])/L10
[7]    CENT←CENT,[1]R[1 3 4]
[8]    →L5
[9]   L10:CENT FAPPEND 43
[10]   'TRACT'
[11]   CENT←0 3⍴⍳0
[12]   CENT←CENT,[1]R[1 3 4]
[13]   KEY←⌊R[1]÷10
[14]  L25:A←TAPE1
[15]   R←CONVC A
[16]   R[1]
[17]   →(99999<R[1])/L50
[18]   →(KEY≠⌊R[1]÷10)/L30
[19]   CENT←CENT,[1]R[1 3 4]
[20]   →L25
[21]  L30:CENT FAPPEND 42
[22]   CENT←0 3⍴⍳0
[23]   CENT←CENT,[1]R[1 3 4]
[24]   KEY←⌊R[1]÷10
[25]   →L25
[26]  L50:CENT FAPPEND 42
[27]   CENT←0 3⍴⍳0
[28]   CENT←CENT,[1]R[1 3 4]
[29]   KEY←⌊R[1]÷1000
[30]  L55:A←TAPE1
[31]   R←CONVC A
[32]   →(' '=↑R)/L65
[33]   →(KEY≠⌊R[1]÷1000)/L60
[34]   CENT←CENT,[1]R[1 3 4]
[35]   →L55
[36]  L60:CENT FAPPEND 41
[37]   CENT←0 3⍴⍳0
[38]   CENT←CENT,[1]R[1 3 4]
[39]   KEY←⌊R[1]÷1000
[40]   →L55
[41]  L65:'END OF BLDCENT'
    ∇
```

```
∇CRCENT1[⎕]∇
    ∇ CRCENT1
[1]    'LEVEL?
[2]    LEV←⎕
[3]    CN←DN←0
[4]   L1:CN←CN+1
[5]    A←FREAD(30+LEV),CN
[6]   L2:DN←DN+1
[7]    B←FREAD(40+LEV),DN
[8]    IDX←(LEV=1)/⌊A[;3]
[9]    IDX←(LEV=2)/⌊A[;3]÷1000
[10]   IDX←(LEV=3)/⌊A[;3]÷10000
[11]   CEN←B[;1]
[12]   →((⌊IDX[1]÷1000)=(⌊CEN[1]÷1000))/L5
[13]   CN,IDX[1],CEN[1]
[14]   ⍞
[15]   →L2
[16] L5:L←IDX∊CEN
[17]   M←CEN∊IDX
[18]   A[L/⍳(1↑⍴A);1 2]←B[M/⍳(1↑⍴B);2 3]
[19]   A←⌊A
[20]   A FREPLACE(30+LEV),CN
[21]   →(CN<( 1+FSIZE(30+LEV)[2]))/L1
    ∇

∇CRCENT2[⎕]∇
    ∇ CRCENT2
[1]    'LEVEL? '
[2]    LEV←⎕
[3]    CN←0
[4]   L1:CN←CN+1
[5]    A←FREAD(30+LEV),CN
[6]    DN← 1+CN
[7]    B←FREAD(40+LEV),DN
[8]    IDX←⌊A[;3]÷1000
[9]    CEN←B[;1]
[10]   →((⌊IDX[1]÷10)=(⌊CEN[1]÷10))/L5
[11]   CN,IDX[1],CEN[1]
[12]   ⍞
[13]   →L1
[14] L5:L←IDX∊CEN
[15]   M←CEN∊IDX
[16]   A[L/⍳(1↑⍴A);1 2]←B[M/⍳(1↑⍴B);2 3]
[17]   A←⌊A
[18]   A FREPLACE(30+LEV),CN
[19]   →(CN<( 1+FSIZE(30+LEV)[2]))/L1
    ∇

∇CRPIC[⎕]∇
    ∇ CRPIC;TAPE1;A;END
[1]    DEFILE
[2]    TIE20
[3]    'IDX2'FTIE 32
[4]    TN←FSIZE 32
[5]    TN←TN[2]-1
[6]    ID←⍳0
[7]    CN←0
[8]    PIC←0 2⍴⍳0
[9]   L1:CN←CN+1
[10]   IDX←⌊(FREAD 32,CN)[;3]÷1000
[11]   ID←ID,[1]IDX
[12]   →(CN<TN)/L1
[13]   CN←0
[14] L4:→((1+⍴ID)<CN←CN+1)/L5
[15]   T←⍳0
[16]   T FAPPEND 22
[17]   T FAPPEND 23
[18]   →L4
[19] L5:CNT←POS←IN←0
[20] L7:A←TAPE1
[21]   CONVP
[22]   →(4≠⍴A)/L15
[23]   A
[24] L10:IDL←A[1]
[25]   IDR←A[2]
[26]   NUM←A[3]
[27]   LEV←A[4]
[28] L15:PIC←PIC,[1]A[⍳2]
[29]   A←TAPE1
[30]   →(0=⍴A)/L25
[31]   CONVP
[32]   →(NUM<IN←IN+1)/L17
[33]   →L15
[34] L17:IN←0
[35]   PIC[;2]←|PIC[;2]
[36]   →(LEV≠1)/L20
[37]   PIC FAPPEND 21
[38]   PIC←0 2⍴⍳0
[39]   CNT←CNT+1
[40]   CNT
[41]   →L7
[42] L20:POS←POS+1
[43]   PIC FAPPEND 24
[44]   PIC←0 2⍴⍳0
[45]   →((IDL∊ID)∨(IDR∊ID))/L22
[46]   COMP←1+⍴ID
[47]   →L24
[48] L22:COMP←ID⍳IDL
[49]   →((COMP=0)∨(COMP>⍴ID))/L23
[50]   IDX←FREAD(LEV+20),COMP
[51]   IDX←IDX,POS
[52]   IDX FREPLACE(LEV+20),COMP
[53] L23:COMP←ID⍳IDR
[54]   →((COMP=0)∨(COMP>⍴ID))/L7
[55] L24:IDX←FREAD(LEV+20),COMP
[56]   IDX←IDX,POS
[57]   IDX FREPLACE(LEV+20),COMP
[58]   →L7
[59] L25:FUNTIE FNUMS
[60]   'END OF BLD PIC'
    ∇
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
∇CRTAB[⎕]∇
    ∇ CRTAB;TAPE1;T
[1]    DEFILE
[2]   ⍝SELECT NUMBER OF DATA
[3]    'NUMBER OF DATA?'
[4]    RL←⎕
[5]   ⍝INITIALIZE WORK AREA AND VARIABLES
[6]    TAT←TAG←TAB←(0,RL)⍴⍳0
[7]    TAT FAPPEND 13
[8]    COMP←CNT←0
[9]    T←TAPE
[10] L4:T←TAPE1
[11]   CONVT
[12]   CNT←CNT+1
[13]   TAB←TAB,[1]⍉T
[14]   KEY←⌊T[1]÷1000
[15]   KEYT←⌊KEY÷10
[16] L5:T←TAPE1
[17]   →(0=⍴T)/L20
[18]   CONVT
[19]   →(KEY=⌊T[1]÷1000)/L15
[20] L10:TG←+⌿0  1↓TAB
[21]   TAG←TAG,[1]TG,KEY×1000
[22]   KEY←⌊T[1]÷1000
[23]   COMP←COMP+1
[24]   TAB FAPPEND 11
[25]   COMP,CNT
[26]   CNT←0
[27]   TAB←(0,RL)⍴⍳0
[28]   →(KEYT=⌊T[1]÷10000)/2+I26
[29]   CRTAB2
[30] L15:TAB←TAB,[1]⍉T
[31]   CNT←CNT+1
[32]   →L5
[33] L20:COMP←COMP+1
[34]   TAB FAPPEND 11
[35]   CRTAB2
[36]   COMP,CNT
[37] L30:'END OF BLDTAB'
    ∇
```

```
∇CRTAB2[⎕]∇
    ∇ CRTAB2
[1]    TAG FAPPEND 12
[2]    TT←+⌿0  1↓TAG
[3]    TT←TT,KEYT×10000
[4]    KEYT←⌊T[1]÷10000
[5]    TAG←(0,RL)⍴⍳0
[6]    TAT←FREAD 13,1
[7]    TAT←TAT,[1]TT
[8]    TAT FREPLACE 13,1
    ∇
     ∇CR10[⎕]∇
    ∇ CR10
[1]    'TAB1'FCREATE 11
[2]    'TAB2'FCREATE 12
[3]    'TAB3'FCREATE 13
    ∇
     ∇CR20[⎕]∇
    ∇ CR20
[1]    'PIC1'FCREATE 21
[2]    'PIC2'FCREATE 22
[3]    'PIC3'FCREATE 23
[4]    'PIC4'FCREATE 24
    ∇
     ∇CR30[⎕]∇
    ∇ CR30
[1]    'IDX1'FCREATE 31
[2]    'IDX2'FCREATE 32
[3]    'IDX3'FCREATE 33
    ∇
     ∇CR40[⎕]∇
    ∇ CR40
[1]    'CENT1'FCREATE 41
[2]    'CENT2'FCREATE 42
[3]    'CENT3'FCREATE 43
```

```
∇ER10[⎕]∇
     ∇ ER10
[1]    'TAB1'FERASE 11
[2]    'TAB2'FERASE 12
[3]    'TAB3'FERASE 13
     ∇
       ∇ER20[⎕]∇
     ∇ ER20
[1]    'PIC1'FERASE 21
[2]    'PIC2'FERASE 22
[3]    'PIC3'FERASE 23
[4]    'PIC4'FERASE 24
     ∇
       ∇ER30[⎕]∇
     ∇ ER30
[1]    'IDX1'FERASE 31
[2]    'IDX2'FERASE 32
[3]    'IDX3'FERASE 33
     ∇
       ∇ER40[⎕]∇
     ∇ ER40
[1]    'CENT1'FERASE 41
[2]    'CENT2'FERASE 42
[3]    'CENT3'FERASE 43
     ∇


∇RESETF[⎕]∇
     ∇ RESETF
[1]    ⎕EX'F'⎕NL 3
[2]    ⎕EX E
     ∇
       ∇RESET [⎕]∇
     ∇ RESETV
[1]    ⎕EX ⍶LPHA ⎕NL 2
     ∇
       ∇TIE10[⎕]∇
     ∇ TIE10
[1]    'TAB1'FTIE 11
[2]    'TAB2'FTIE 12
[3]    'TAB3'FTIE 13
     ∇
       ∇TIE20[⎕]∇
     ∇ TIE20
[1]    'PIC1'FTIE 21
[2]    'PIC2'FTIE 22
[3]    'PIC3'FTIE 23
[4]    'PIC4'FTIE 24
     ∇
       ∇TIE30[⎕]∇
     ∇ TIE30
[1]    'IDX1'FTIE 31
[2]    'IDX2'FTIE 32
[3]    'IDX3'FTIE 33
     ∇
       ∇TIE40[⎕]∇
     ∇ TIE40
[1]    'CENT1'FTIE 41
[2]    'CENT2'FTIE 42
[3]    'CENT3'FTIE 43
     ∇


∇DEFILE[⎕]∇
     ∇ DEFILE;RL;BS
[1]    'LOGICAL RECORD LENGTH?
[2]    RL←⎕
[3]    'BLOCK SIZE? '
[4]    BS←⎕
[5]    ⎕CTS←'FILEDEF TAPE1 CLEAR'
[6]    DCB←RL,' BLOCK ',BS,' RECFM FB 9TRACK DEN 800)'
[7]    ⎕CTS←'FILEDEF TAPE1 TAP1 (LRECL ',DCB
[8]    ⎕CTS←'TAPE REW'
[9]    ⎕CTS←'TAPE FSF'
[10]   TAPE1←'TAPE1 (370'
[11]   (2=111 ⎕SVO'TAPE1')/' TAPE1 DEFINED'
     ∇
       ∇MESSAGE[⎕]∇
     ∇ MESSAGE TAPEID
[1]    ⎕CTS←'MSG OP PLEASE MOUNT TAPE ',TAPEID,' AS TAP1 9TRACK 800BPI
[2]    ⎕CTS←'MOUNT ',TAPEID
[3]    'SENT'
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
∇CRIDX[⎕]∇
    ∇ CRIDX
[1]    TIE10
[2]    TIE30
[3]    IDX2←IDX3←0 5⍴⍳0
[4]    CNTB←CNTG←0
[5]    IN←1
[6]    GN← 1+(FSIZE 11)[2]
[7]    IDX←(FREAD 11,IN)[;56]
[8]    KEY←⌊IDX[1]÷10000
[9]    →L7
[10] L5:IN←IN+1
[11]   'INREC= ',⍕IN
[12]   →(IN>GN)/L20
[13]   IDX←(FREAD 11,IN)[;56]
[14]   →(KEY≠⌊IDX[1]÷10000)/L10
[15] L7:IDX1←((⍴IDX),3)⍴0
[16]   IDX1[;3]←IDX
[17]   IDX1 FAPPEND 31
[18]   CNTB←CNTB+1
[19]   X2←5⍴0
[20]   X2[3]←1000×IG←⌊IDX[1]÷1000
[21]   X2[4]←CNTB
[22]   X2[5]←1↑⍴IDX1
[23]   IDX2←IDX2,[1]X2
[24]   →L5
[25] L10:IDX2 FAPPEND 32
[26]   CNTG←CNTG+1
[27]   CNTG
[28]   X3←5⍴0
[29]   X3[3]←KEY×10000
[30]   X3[4]←CNTG
[31]   X3[5]←1↑⍴IDX2
[32]   IDX3←IDX3,[1]X3
[33]   KEY←⌊IDX[1]÷10000
[34]   IDX2←0 5⍴⍳0
[35]   →L7
[36] L20:IDX3 FAPPEND 33
[37]   1↑⍴IDX3
[38]   'END OF BLDIDX'
    ∇
```

APPENDIX E

LUMIS INTERACTIVE GRAPHICS FORTRAN PROGRAM LISTINGS

```
      DIMENSION         IAGG(12), IDAT(233)
   10 FORMAT(2I1)
   20 FORMAT(108X,I4,I2,I3,I2,1X,12I12,233I6,18X)
   30 FORMAT(I10,12I12,233I6)
   40 FORMAT(1H ,'READ ERROR AFTER REC NO = ',I5)
   50 FORMAT(1H0,'ERROR COUNT = ',I10//'OUTPUT COUNT = ',I10)
      ICNT=0
      IERR=0
  100 READ(2,20,ERR=400,END=500)
     1     ID1,ID2,ID3,ID4,(IAGG(J),J=1,12),(IDAT(K),K=1,233)
      IDD=((ID1  *10+ID2  )*10000)+(ID3  *10+ID4)
      WRITE(3,30) IDD,(IAGG(J),J=1,12),(IDAT(K),K=1,233)
      ICNT=ICNT+1
      GO TO 100
  400 IERR=IERR+1
      WRITE(6,40) ICNT
      GO TO 100
  500 WRITE(6,50) IERR,ICNT
      STOP
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DIMENSION IXY(4)
      IN=2
      IO=3
      IFLG=0
      ICNTO=0
      IERR=0
      IERR1=0
      IERR2=0
      IBNKL=0
      IBNKR=0
   10 FORMAT(55X,I1,28X,I6,1X,I4,I2,I4,I2,48X,I3,I2,14X,I3,I2,55X,4I7)
   20 FORMAT(7I10)
   30 FORMAT(1H1,15HRECORD COUNTS  /
     -        1H0,15HRECORDS READ   ,I10/
     -        1H0,15HRECORDS WRITTEN,I10/
     -        1H0,15HLEFT BLANK     ,I10/
     -        1H0,15HRIGHT BLANK    ,I10/
     -        1H0,15HCODING FLAG2   ,I10/
     -        1H0,15HCODING FLAG1   ,I10/
     -        1H0,15HOTHER ERROR    ,I10)
   40 FORMAT(1H ,7I10)
   50 FORMAT(1H ,'WRITE ERROR REC NO = ',I5)
  100 DO 850 KK=1,21000
      READ(IN,10,END=900) NCLF,NREC,NTL,NTLS,NTR,NTRS,
     -                    NBL,NBLS,NBR,NBRS,(IXY(I),I=1,4)
      NTL=NTL*10+NTLS
      NTR=NTR*10+NTRS
      NBL=NBL*10+NBLS
      NBR=NBR*10+NBRS
      IF(NCLF .GT. 0 .OR. NTR .EQ. 0 .OR. NTL .EQ. 0) GO TO 600
      IF(NTL .EQ. NTR) GO TO 200
      LEV=4
      GO TO 500
  200 NGL=NBL/1000
      NGR=NBR/1000
      IF(NGL .EQ. NGR) GO TO 300
      LEV=3
      GO TO 500
  300 IF(NBL .EQ. NBR) GO TO 400
      LEV=2
      GO TO 500
  400 LEV=1
      GO TO 550
  500 ID =NTR*10000+NBR
      IFLG=1
      GO TO 820
  550 ID =NTL*10000+NBL
      GO TO 820
  600 LEV=4
      IF(NCLF .EQ. 2) IERR2=IERR2+1
      IF(NCLF .EQ. 1) IERR1=IERR1+1
      IF(NTR .EQ. 0 .AND. NTL .NE. 0) GO TO 700
      IF(NTL .EQ. 0 .AND. NTR .NE. 0) GO TO 800
      IERR=IERR+1
      GO TO 550
  700 IBNKR=IBNKR +1
      ID =NTL*10000+NBL
      GO TO 820
  800 IBNKL=IBNKL+1
      ID =NTR*10000+NBR
  820 WRITE(IO,20,ERR=830) (IXY(I),I=1,4),ID,NREC,LEV
C     WRITE( 6,40) (IXY(I),I=1,4),ID, NREC,LEV
      ICNTO=ICNTO+1
      IF(IFLG .NE. 1) GO TO 850
      IFLG=0
      ID =NTL*10000+NBL
      GO TO 820
  830 WRITE(6,50) KK
  850 CONTINUE
  900 END FILE 3
      WRITE(6, 30)    KK,ICNTO,IBNKL,IBNKR,IERR2,IERR1,IERR
      STOP
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IMPLICIT INTEGER (A-Z)
      NINES=999999999
      IN=8
      OUT=9
      RECIN=0
      RECOUT=0
C
C
C   READ DIME FILE RECORD AND SELECT REQUIRED INFORMATION
C
  100 READ(IN,10,END=800)
     1RN,TL,TLS,TR,TRS,FN,TN,BL,BLS,BR,BRS,FY,FX,TY,TX
      RECIN=RECIN+1
C
C     MAKE BLOCK NUMBERS UNIQUE BY COMBINING WITH TRACT NUMBERS,
C
      BLKL=0
      GRPL=0
      TRTL=0
      BLKR=0
      GRPR=0
      TRTR=0
      IF(TL .EQ. 0) GO TO 200
      BLKL=(TL*10+TLS)*10000+(BL*10+BLS)
      GRPL=BLKL/1000
      TRTL=GRPL/10
  200 IF(TR .EQ. 0) GO TO 300
      BLKR=(TR*10+TRS)*10000+(BR*10+BRS)
      GRPR=BLKR/1000
      TRTR=GRPR/10
  300 LEV=1
C
C     SELECT ONLY BOUNDARY SEGMENTS.
C
  350 IF(BLKL .EQ. BLKR) GO TO 100
      IF(GRPL .NE. GRPR) LEV=2
      IF(TRTL .NE. TRTR) LEV=3
C
      IF(GRPL .EQ. 0) GO TO 400
      WRITE(OUT,20) GRPL,GRPR,RN,FN,TN,FX,FY,TX,TY,LEV
      RECOUT=RECOUT+1
      GO TO 100
C
  400 WRITE(OUT,20) GRPR,GRPL,RN,TN,FN,TX,TY,FX,FY,LEV
      RECOUT=RECOUT+1
  500 GO TO 100
  800 WRITE(OUT,30) NINES
      END FILE OUT
      WRITE(6,40)  RECIN, RECOUT
   10 FORMAT(84X,I6,1X,2(I4,I2),19X,2I4,21X,I3,I2,14X,I3,I2,55X,4I7)
   20 FORMAT(2I9,3I6,4I9,I8)
   30 FORMAT(I9)
   40 FORMAT('0  PRESORT PROCESSING COMPLETED' /// 10X, I6, ' RECORDS IN
     X' // 10X, I6, ' RECORDS PASSED TO SORT')
      CALL  EXIT
      STOP
      END
```

```
      IMPLICIT INTEGER (B-Y)                                            DAC00030
      INTEGER*2 ND(4)                                                   DAC00040
      DIMENSION NQ(2),NS(3)                                             DAC00050
      EQUIVALENCE (NQ(1),ND(1))                                         DAC00060
      DOUBLE PRECISION ZMULT                                            DAC00070
      DIMENSION TITLE(3), OPTION(3)                                     DAC00080
       DIMENSION  TLIST(2000)                                           DAC00090
      COMMON      ELIST( 2000, 5 ) , ZMULT                              DAC00100
C                                                                       DAC00110
      DATA  R,  INFILE, OUT1, OUT2,     EOF   , MAX , AREA, ATOTAL, ASUB DAC00120
     X /   1,     9,     10,    11 , 999999999,2000 , 0.0 ,  0.0,'  0.0 /
      LOGICAL PIOS / .FALSE. /                                          DAC00140
       LOGICAL  ADJ / .FALSE. /                                         DAC00150
      LOGICAL   LIST  / .FALSE. / , BNDS  / .FALSE. /                   DAC00160
      DATA   NUM  / 0 / ,  BGRP / 'BGPS' /                              DAC00170
      DATA  BLKS  / 'BLKS' /  , TDRPT /0/, DROPT /0/                    DAC00180
      DATA BLANK /' '/ , IA /'A'/                                       DAC00190
C                                                   *                   DAC00210
C                                             OPTION  PROCESSING        DAC00220
C                                                                       DAC00230
      CNT=0
      READ(5,5001,ERR=9999) AMULT,UNITS,KALC,OPTION,TITLE
 5001 FORMAT(F10.0,2A4,3I1,3A4)
C                                                                       DAC00270
      ZMULT = AMULT * .00000001                                         DAC00300
      IF (OPTION(1) .EQ. 1) LIST = .TRUE.                               DAC00310
      IF (OPTION(2) .EQ. 1) BNDS = .TRUE.                               DAC00320
      IF (OPTION(3) .EQ. 1) ADJ  = .TRUE.                               DAC00330
       READ( INFILE, 5002, END=500) KEY, TLIST(1), (ELIST(1,J),J=1,5)
 5002 FORMAT(7I9)
      WRITE(6,5014) TITLE, KALC, UNITS, OPTION, KEY                     DAC00410
 5014 FORMAT( '1', 10X, 'U. S. BUREAU OF CENSUS  -  DIME AREA-CENTROID SDAC00420
     XYSTEM (DACS)' //  25X, 3A4 // 11X, 'CALCULATIONS ARE FOR ',A4,    DAC00430
     X ' IN ',A4,5X,3I1,' IS OPTION'/// 5X, 'FIRST AREA IS',I11  // )   DAC00440
  100 R=R+1                                                             DAC00450
      IF( R .LT. MAX )  GO TO 120                                       DAC00460
      WRITE(6,5003) KALC, KEY, MAX                                      DAC00470
 5003 FORMAT( // ' *** ', A4, I11, ' HAS',I5, ' OR MORE SEGMENTS' )     DAC00480
      DROPT = DROPT + 1                                                 DAC00490
      NIX = -1                                                          DAC00500
  105  READ( INFILE, 5002, END=500)KEYX, TLIST(1), (ELIST(1,J),J=1,5)
      NIX = NIX + 1                                                     DAC00520
      IF( KEYX .EQ. KEY ) GO TO 105                                     DAC00530
      WRITE( 6, 5004 )  NIX                                             DAC00540
 5004 FORMAT( 24X, I5, ' SEGMENTS DROPPED.  NO  CALCULATIONS' // )      DAC00550
      NIX  =  0                                                         DAC00560
      R = 2                                                             DAC00570
C                                                                       DAC00580
C                                                                       DAC00590
  120  READ( INFILE, 5002, END=500)KEYX, TLIST(R), (ELIST(R,J),J=1,5)
C                                                                       DAC00620
      IF( KEYX .EQ. KEY ) GO TO 100                                     DAC00630
C                                                                       DAC00640
C  $                *                                                   DAC00650
C  $  PROCESSING    *                                                   DAC00660
C  $                *                                                   DAC00670
C                                                                       DAC00680
  130 NUM = NUM + 1                                                     DAC00690
C                                                                       DAC00700
      NBLS   = R-1                                                      DAC00710
      AREA   = 0.0                                                      DAC00720
      CENTX  = 0                                                        DAC00730
      CENTY  = 0                                                        DAC00740
      IND    = 0                                                        DAC00750
      CALL  CHAIN (  NBLS, COMPS, CLOSES, RVSLS )                       DAC00760
      IF( COMPS .NE. CLOSES ) GO TO 140                                 DAC00770
      IF( COMPS .EQ.   1      ) GO TO 150                               DAC00780
C                                                                       DAC00790
      WRITE( 6,5005) KALC, KEY, COMPS                                   DAC00800
C                                                                       DAC00810
 5005 FORMAT( '0*** ',A4, I11, ' HAS', I3, ' BOUNDED REGIONS - CHECK' ) DAC00820
      GO TO 150                                                         DAC00830
C                                                                       DAC00840
  140 WRITE( 6,5006) KALC, KEY, COMPS, CLOSES                           DAC00850
 5006 FORMAT( '0*** ',A4, I11, ' HAS', I3, ' COMPONENTS.  ONLY',I3,' ARE
     X CLOSED.  REGION  NOT PROCESSED' )                                DAC00870
C                                                                       DAC00880
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IMPLICIT INTEGER (A-Z)
      DIMENSION REC(200,10),VTX(500,2)
   10 FORMAT(2I9,3I6,4I9,I8)
   20 FORMAT(4I9)
   30 FORMAT(1H0,4I10)
   40 FORMAT(2I9)
   45 FORMAT(1H ,2I9)
   50 FORMAT(1H1,'VERTEX FILE SUMMARY'//
     1            4X,'IDL',7X,'IDR',7X,'VTX',10X,'LEV')
   60 FORMAT(1H0,'NO OF CHAIN = ',I6,10X,'NO OF REC = ',I6)
   70 FORMAT(1H0,'WRITE ERROR')
      NINES=999999999
      IN=8
      OUT=9
      R=1
      NUM=0
      NREC=0
      READ(IN,10,END=900) (REC(R,J),J=1,10)
      IDL=REC(R,1)
      IDR=REC(R,2)
  200 R=R+1
      READ(IN,10,END=900) (REC(R,J),J=1,10)
      IF(IDL .EQ. REC(R,1) .AND. IDR .EQ. REC(R,2)) GO TO 200
      CNT=R-1
      CHK=0
      LL=0
      K=0
  300 K=K+1
      FLG=0
      IF(K .GT. CNT) GO TO 400
      M=K
      IF(REC(M,3) .EQ. 0) GO TO 300
      LL=LL+1
      VTX(LL,1)=(-1)*REC(M,7)
      VTX(LL,2)=REC(M,6)
      KEY1=REC(M,9)
      KEY2=REC(M,8)
      REC(M,3)=0
      N=0
  320 N=N+1
      IF(N .GT. CNT) GO TO 350
      IF(REC(N,3) .EQ. 0) GO TO 320
      MAT=0
      IF(KEY1 .EQ. REC(N,7) .AND. KEY2 .EQ. REC(N,6)) GO TO 370
      MAT=1
      IF(KEY1 .EQ. REC(N,9) .AND. KEY2 .EQ. REC(N,8)) GO TO 370
      GO TO 320
  350 FLG=1
  370 LL=LL+1
      VTX(LL,1)=KEY1
      VTX(LL,2)=KEY2
      CHK=CHK+1
      IF(CHK .EQ. CNT .OR. CHK .GT. CNT) GO TO 400
      IF(FLG .EQ. 1) GO TO 300
      IF(MAT .EQ. 1) GO TO 380
      KEY1=REC(N,9)
      KEY2=REC(N,8)
      GO TO 390
  380 KEY1=REC(N,7)
      KEY2=REC(N,6)
  390 REC(N,3)=0
      M=N
      N=0
      GO TO 320
  400 WRITE(OUT,20,ERR=800) IDL,IDR,LL,REC(1,10)
      DO 410 I=1,LL
      WRITE(OUT,40,ERR=800) (VTX(I,J),J=1,2)
  410 CONTINUE
      WRITE(OUT,40) NINES
      NUM=NUM+1
      NREC=NREC+LL
      DO 420 J=1,10
      REC(1,J)=REC(R,J)
  420 CONTINUE
      IDL=REC(R,1)
      IDR=REC(R,2)
      R=1
      GO TO 200
  800 WRITE(6,70)
  900 END FILE OUT
      WRITE(6,60) NUM,NREC
      STOP
      END
```

```
      WRITE( 6, 5007 )                                                  DAC00890
      DO 145 I = 1,NBLS                                                 DAC00900
      NQ(1) = ELIST(I,2)                                                DAC00910
      NQ(2) = ELIST(I,3)                                                DAC00920
      NS(1) = ELIST(I,1)                                                DAC00930
      NS(2) = ELIST(I,4)                                                DAC00940
      NS(3) = ELIST(I,5)                                                DAC00950
145   WRITE(6,5097)  ND, NS                                             DAC00960
5097  FORMAT(2(5X,A2,I5),3I12)                                          DAC00970
C                                                                       DAC00980
 5007 FORMAT(  '0   SEGMENT  LISTING: '       //                       'DAC00990
     X   FROM NODE      TO NODE  REC NUMBER     FROM X       FROM Y' /)
C                                                                       DAC01010
      DROPT = DROPT + 1                                                 DAC01020
C                                                                       DAC01030
      GO TO 1000
C                                                                       DAC01050
  150 CALL  CALC( NBLS, AREA, CENTX, CENTY , MX, MY )                   DAC01060
C                                                                       DAC01070
      IF( AREA .LE. 0.0 ) DROPT = DROPT + 1                             DAC01080
C                                                                       DAC01090
      IF( CENTX .NE. 0. ) CALL POLYPT( NBLS, CENTX, CENTY, IND,MX,MY)   DAC01100
      IF(.NOT. BNDS ) GO TO 1000                                        DAC01110
C                                                                       DAC01120
      SYMAP=IA                                                          DAC01130
      KEY1=KEY/10000                                                    DAC01140
      KEY2=MOD(KEY,10000)                                               DAC01150
      KEY2=KEY2*100                                                     DAC01160
             DO  180  I = 1, NBLS                                       DAC01170
      KEY2=KEY2+I                                                       DAC01180
      WRITE(OUT2,5011) KEY1,SYMAP,ELIST(I,5),ELIST(I,4),ELIST(I,1),     DAC01190
     XKEY2                                                              DAC01200
 5011 FORMAT(I5,4X,A1,2I10,20X,I10,13X,I7)                              DAC01210
      SYMAP=BLANK                                                       DAC01220
  180 CONTINUE                                                          DAC01230
C                                                                       DAC01240
      KEY2=KEY2+1                                                       DAC01250
      WRITE(OUT2,5011) KEY1,SYMAP,ELIST(1,5),ELIST(1,4),ELIST(1,1),     DAC01260
     XKEY2                                                              DAC01270
C                                                                       DAC01280
 1000 IF( LIST ) WRITE(6,5008)KALC, KEY, NBLS, AREA, UNITS, CENTX,CENTY DAC01290
 5008 FORMAT( //5X, A4, I11, I9, ' SEGMENTS.  AREA =', F15.5,1X, A4     DAC01390
     X          /        29X    ,     '    CENTROID  IS :',2I15      )  DAC01400
      IF (PIOS) CALL REFMT (NBLS,KEY)                                   DAC01300
C                                                                       DAC01310
      IF(RVSLS .GT. 0) WRITE(6,5015) RVSLS
 5015 FORMAT( 9X, I11, '  REVERSALS  - CHECK FOR POSS. ERRORS.' )       DAC01340
      IF( IND .GE. 0 )  GO TO 1100                                      DAC01330
C                                                                       DAC01380
      CALL ADJUST ( NBLS, CENTX, CENTY, NODE )                          DAC01350
C                                                                       DAC01360
                      WRITE(6,5009) NODE , CENTX, CENTY
 5009 FORMAT( / 9X, 'CENTROID WAS OUTSIDE  BOUNDARY - ADJUSTED TO NODE',DAC01420
     X  I11    /        29X     ,      'NEW CENTROID  IS :',2I15     )  DAC01430
C                                                                       DAC01410
 1100 WRITE( OUT1, 5010 ) KEY, AREA, CENTX, CENTY                       DAC01440
C                                                                       DAC01450
 5010 FORMAT( I10, F20.5, 2I10 )                                        DAC01460
C                                                                       DAC01470
      ASUB  =  ASUB + AREA                                              DAC01480
      ATOTAL=  ATOTAL + AREA                                            DAC01490
C                                                                       DAC01500
 1150  IF( ADJ  )  CALL ADJNCY ( NBLS, KEY , TLIST )                    DAC01510
      DO 1200 I=1,5
 1200 ELIST(1,I)= ELIST(R,I)                                            DAC01530
C                                                                       DAC01540
 1250 KEY = KEYX                                                        DAC01550
      R = 2                                                             DAC01560
C                                                                       DAC01720
C                                                                       DAC01730
 1300 IF( KEY .LT. EOF ) GO TO 120                                      DAC01740
C                                                                       DAC01750
      IF( TDRPT .EQ. 0 ) TDRPT = DROPT                                  DAC01760
      WRITE(6,5013)  NUM, KALC, TDRPT, ATOTAL, UNITS                    DAC01770
 5013 FORMAT( '1'  / 1X, 10('-----') // I10, 1X, A4, ', ', I6, ' OMITTED'DAC01780
     X          // 5X, 'TOTAL AREA IS' , F13.5, 1X, A4 // 1X, 10('-----') ) DAC01790
```

```
      GO TO 1350
  500 KEYX= EOF                                                         DAC01810
      GO TO 1250                                                        DAC01820
C                                                                       DAC01830
 9999 WRITE(6,5099)                                                     DAC01840
C                                                                       DAC01850
 5099 FORMAT( '1  ERROR  IN  FIRST  RECORD' )                           DAC01860
      WRITE(6,5001)I, AMULT, UNITS, KALC, OPTION, TITLE                 DAC01870
 1350 STOP
      END                                                               DAC01890




      SUBROUTINE CHAIN (NBLS,COMPS,CLOSES,RVSLS)
C                                                                       DAC02200
C                                                                       DAC02210
C    THIS SUBROUTINE IS IDENTICAL TO THAT USED  IN THE DIME EDIT PACKAGEDAC02220
C                                                                       DAC02230
C                                                                       DAC02240
C                                                                       DAC02250
C                                                                       DAC02260
      IMPLICIT INTEGER (A-Z)                                            DAC02270
      DOUBLE PRECISION  ZMULT                                           DAC02280
      DIMENSION HOLDER(5),ELIST(2000,5)                                 DAC02290
      COMMON ELIST,ZMULT                                                DAC02300
      BEGIN  = 1                                                        DAC02310
      END    = 1                                                        DAC02320
      COMPS  = 0                                                        DAC02330
      RVSLS  = 0                                                        DAC02340
      CLOSES = 0                                                        DAC02350
      HEAD = ELIST(BEGIN,2)                                             DAC02360
      TAIL = ELIST(END,3)                                               DAC02370
      IF( NBLS - 1 )  1250, 1200, 1000                                  DAC02380
 1000 IF( HEAD .EQ. TAIL )  GO TO 1200                                  DAC02390
      START = END + 1                                                   DAC02400
      IF( START .GT. NBLS )  GO TO 1200                                 DAC02410
            DO 1100 I = START, NBLS                                     DAC02420
      IF (ELIST(I,3).EQ.HEAD) GO TO 3000                                DAC02430
      IF (ELIST(I,2).EQ.TAIL) GO TO 2000                                DAC02440
 1100       CONTINUE                                                    DAC02450
            DO 1150 I = START, NBLS                                     DAC02460
      I1 = I + 1                                                        DAC02470
         IF (I.EQ.NBLS) I1 = 1                                          DAC02480
         IF (ELIST(I,2).EQ.HEAD) GO TO 2990                             DAC02490
      IF (ELIST(I,3).EQ.TAIL) GO TO 1990                                DAC02500
 1150       CONTINUE                                                    DAC02510
 1200 COMPS = COMPS + 1                                                 DAC02520
      IF( HEAD .EQ. TAIL )        CLOSES = CLOSES + 1                   DAC02530
 1250 IF( END .GE. NBLS )  RETURN                                       DAC02540
CHAIN **                                                                DAC02550
            ELIST(END,1) = -ELIST(END,1)                                DAC02560
      IF (CLOSES.GE.2) GO TO 1400                                       DAC02570
C                                                                       DAC02580
C  THE FOLLOWING SECTION REARRANGES THE ORDER OF SEGMENTS IN THE CHAIN TDAC02590
C       PERMIT CHAINING AS ONE COMPONENT FOR FIGURE EIGHT OR CHECKERBOARDAC02600
C       CONFIGURATIONS, VIZ.                                            DAC02610
C                                                                       DAC02620
C              XXXXXXXX             XXXX        XXX                     DAC02630
C              X      X             X   X     X   X   XXXXXXXX          DAC02640
C              X      X              X   X   X     X X       X          DAC02650
C              XXXXXXXXXXXXXXXXX      X   XX X      X        X          DAC02660
C                     X        X      X    X       X XXXXXXXXX          DAC02670
C                     X        X        XXXX XXX   X                    DAC02680
C                     X        X                XXX                     DAC02690
C                     XXXXXXXXXX                                        DAC02700
C                                                                       DAC02710
C      ARE TYPES OF REGIONS WHICH CONTAIN ONLY ONE COMPONENT BUT WHICH TDAC02720
C      THE PROGRAM CAN INTERPRET AS HAVING TWO OR THREE CLOSED COMPONENTDAC02730
C                                                                       DAC02740
      CLOSES = 0                                                        DAC02750
      COMPS = 0                                                         DAC02760
      J1 = END + 1                                                      DAC02770
      END1 = END                                                        DAC02780
```

```
      DO 1300 K1 = 1,END1
      HEAD = ELIST(K1,3)
      DO 1300 I = J1,NBLS
      IF (ELIST(K1,3).EQ.ELIST(I ,2)) GO TO 2000
 1300 CONTINUE
      DO 1350 K1 = 1,END1
      HEAD = ELIST(K1,2)
      DO 1350 I = J1,NBLS
      I1 = I + 1
      IF (I.EQ.NBLS) I1 = 1
      IF (ELIST(K1,3).EQ.ELIST(I ,3)) GO TO 1990
 1350 CONTINUE
      CLOSES = 1
      COMPS = 1
C
C
 1400 END = END + 1
      BEGIN = END
      HEAD = ELIST(BEGIN,2)
      TAIL = ELIST(END,3)
      GO TO 1000
 1990 TEMP = ELIST(I,2)
      ELIST(I,2) = ELIST(I,3)
      ELIST(I,3) = TEMP
      TEMPY = ELIST(I,5)
      TEMPX = ELIST(I,4)
      ELIST (I,4) = ELIST (I1,4)
      ELIST (I,5) = ELIST (I1,5)
      ELIST(I1,4) = TEMPX
       ELIST(I1,5) = TEMPY
      RVSLS      =  RVSLS + 1
 2000 END = END + 1
      IF( END .EQ. I )  GO TO 2050
      DO 2010  K=1,5
 2010 HOLDER(K) = ELIST(I,K)
      TEMP = I
 2020 TEMP = TEMP - 1
      DO 2025 K=1,5
 2025 ELIST(TEMP+1,K) = ELIST(TEMP,K)
      IF( TEMP .GT. END )  GO TO 2020
      DO 2030 K=1,5
 2030 ELIST(END,K) = HOLDER(K)
 2050 TAIL = ELIST(END,3)
      GO TO 1000
 2990 TEMP = ELIST(I,2)
      ELIST(I,2) = ELIST(I,3)
      ELIST(I,3) = TEMP
      TEMPY = ELIST(I,5)
      TEMPX = ELIST(I,4)
      ELIST (I,4) = ELIST (I1,4)
      ELIST (I,5) = ELIST (I1,5)
      ELIST(I1,4) = TEMPX
      ELIST(I1,5) = TEMPY
      RVSLS      =  RVSLS + 1
 3000 END = END + 1
      DO 3010  K=1,5
 3010 HOLDER(K) = ELIST(END,K)
      TEMP = END
 3020 TEMP = TEMP - 1
      DO 3030 K=1,5
 3030 ELIST(TEMP+1,K) = ELIST(TEMP,K)
      IF( TEMP .GT. BEGIN )  GO TO 3020
      IF( I .EQ. END )  I = BEGIN
      DO 3100 K=1,5
      ELIST(BEGIN,K) = ELIST(I,K)
 3100 ELIST(I,K) = HOLDER (K)
      HEAD = ELIST (BEGIN,2)
      GO TO 1000
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE        ADJUST( NBLS, CX, CY, NODE )
C
C        ROUTINE TO ADJUST BAD CENTROIDS
C        TO NEAREST NODE
C
      IMPLICIT  INTEGER (B-Y)
      DOUBLE PRECISION  ZMULT
      COMMON  ELIST( 2000, 5 ),  ZMULT
C
      ND  =   0
      DO 101 I = 1,NBLS
      DX  =  CX-ELIST(I,4)
      DY  =  CY-ELIST(I,5)
      AXY =  DX * DX  +  DY * DY
      IF( I .EQ. 1 ) GO TO 100
      IF( A  .LT.  AXY )  GO TO 101
  100 A = AXY
      ND  =  I
  101 CONTINUE
      IF( ND  .LE.  0 )  GO TO 200
      CX =  ELIST(ND,4)
      CY =  ELIST(ND,5)
      NODE = ELIST(ND,2)
      RETURN
C
  200 CX =  0
      CY =  0
      NODE =0
      RETURN
      END
```

```
            SUBROUTINE  CALC( NSEG, AREA, CX, CY, MINX, MINY )
C
      IMPLICIT INTEGER ( B-Y )
      DIMENSION   ELIST(2000,5)
      COMMON  ELIST,  ZMULT
      DOUBLE PRECISION  ZMULT, ARSIX, A, DX,DY, X1,Y1, X2,Y2
      DIMENSION  ARS(10)
      BEGIN = 1
      NAR = 0
      A = 0
      DX = 0
      DY = 0
C
C
C       USE  FIRST  COORDINATES  TO REDUCE  ALL  OTHERS
C
C
      MINX = ( ELIST(1,4) / 10000   ) * 10000
      MINY = ( ELIST(1,5) / 10000   ) * 10000
C                                CHECK FOR MULTIPLE BOUNDARIES
    5 NAR = NAR + 1
      DO 20 I = BEGIN, NSEG
      IF( ELIST(I,1). ) 10,20,20
   10 END = I
      GO TO 30
   20 CONTINUE
      END = NSEG
C
C
C       REDUCING  COORDINATES  AND USE OF  DOUBLE PRECISION
C       MINIMIZES TRUNCATION AND ROUNDING ERRORS.
C
C
   30   X1 = ELIST(END,4) - MINX
        Y1 = ELIST(END,5) - MINY
      DO 100 K = BEGIN,END
        X2 =  ELIST( K,4) - MINX
        Y2 =  ELIST( K,5) -MINY
```

```
C
      A  =  A +  (X2-X1) * (Y1+Y2)
C
        X1 = X2
        Y1 = Y2
C
  100 CONTINUE
C                             ACCUMULATE  AREAS IN 'ARS'
C
      ARS(NAR) = -A
       A = 0.0000000
C                        COMPUTE  CENTROID OF LARGEST AREA ONLY
      BEGIN = END + 1
      IF( END .LT. NSEG ) GO TO 5
      BEGIN = 1
      AB = ARS(1)
      IAB= 1
      A = AB
            IF (NAR.EQ.1) GO TO 180
      IF(NAR .GT. 10) GO TO 500
             DO 150 I = 2,NAR
             A = A + ARS(I)
             IF( ARS(I) .LE. AB ) GO TO 150
                                                 AB = ARS(I)
                                                IAB = I
  150        CONTINUE
  180 AREA = A * ZMULT
      IF( A .EQ. 0.0 ) GO TO 500
      IF(NAR .EQ.  1 ) GO TO 200
       K = 1
             DO 185 I = 1, NSEG
             IF( ELIST(I,1) .GE. 0 ) GO TO 185
             K = K + 1
             IF( K  -   IAB )185,183,184
  183        BEGIN = I + 1
             GO TO 185
  184        END = I
             A = AB
             GO TO 200
  185        CONTINUE
             END = NSEG
             A = AB
  200 X1 =   ELIST(END,4) - MINX
      Y1 =   ELIST(END,5) - MINY
      ARSIX = A * 3.0
C
C     NOTE  THAT   X1 AND Y1  ARE ALREADY SET
C
       DO 300 K = BEGIN,END
       X2   =  ELIST( K,4) -   MINX
       Y2   =  ELIST( K,5) -   MINY
C
      DX    =  DX + ( Y2 - Y1 ) * (X1*X1 + X1*X2 + X2*X2 ) / ARSIX
      DY    =  DY + ( X2 - X1 ) * (Y1*Y1 + Y1*Y2 + Y2*Y2 ) / ARSIX
C
       X1   =  X2
       Y1   =  Y2
C
  300 CONTINUE
C
      CX    =  DX + MINX
      CY =  -DY + MINY
C
C
C
C
C
      RETURN
  500 CX    =    0
      CY    =    0
      RETURN
      END
```

```
      SUBROUTINE         INSECT       ( IXY, IND )
C     SUBROUTINE  TO  CHECK  TWO LINE SEGMENTS    FOR INTERSECTION      DAC05610
C                                                                       DAC05620
C     THE ENDPOINTS OF THE LINES ARE TRANSMITTED (INTEGER BINARY)       DAC05630
C     IN THE ARRAY 'IXY'  (FIRST  8  ELEMENTS) ( X-Y X-Y  X-Y X-Y   )   DAC05640
C                                                                       DAC05650
C     THE VARIABLE 'IND' IS RETURNED:  0  IF NO INTERSECTION            DAC05660
C                                     -1  IF LINES ARE COINCIDENT       DAC05670
C                                     +1  IF THEY INTERSECT             DAC05680
C                                    THE COORDINATES OF INTERSECTION    DAC05690
C                            ARE RETURNED IN IXY(9) AND IXY(10)         DAC05700
C                                                                       DAC05710
C                                                                       DAC05720
      DIMENSION IXY (10),XY(8),S(2),P(2)                                DAC05730
      IND = 0                                                           DAC05740
      X = 0                                                             DAC05750
      Y = 0                                                             DAC05760
C                                                                       DAC05770
C     RETURN IF NO INTERSECTION POSSIBLE                                DAC05780
C                                                                       DAC05790
C                                       CROSS PRODUCT CALCULATION       DAC05800
       ACX = IXY(1) - IXY(5)                                            DAC05810
       ACY = IXY(2) - IXY(6)                                            DAC05820
       ADX = IXY(1) - IXY(7)                                            DAC05830
       ADY = IXY(2) - IXY(8)                                            DAC05840
C                                                                       DAC05850
       BCX = IXY(3) - IXY(5)                                            DAC05860
       BCY = IXY(4) - IXY(6)                                            DAC05870
       BDX = IXY(3) - IXY(7)                                            DAC05880
       BDY = IXY(4) - IXY(8)                                            DAC05890
      A1 =  (ACX * ADY - ACY * ADX)                                     DAC05900
      A2 =                              (BCX * BDY - BCY * BDX)         DAC05910
       IF  ( A1 * A2  .GT. 0.0 )RETURN                                  DAC05920
       R =  (ACX * BCY - ACY * BCX) * (ADX * BDY - ADY * BDX)           DAC05930
       IF  ( R  .GT. 0.0)RETURN                                         DAC05940
C  CHECK COLINEARITY                                                    DAC05950
      IF(A1.EQ.0 .AND. A2.EQ. 0) GO TO 2220                             DAC05960
C  CALCULATE INTERSECTION                                               DAC05970
      R = 0.0                                                           DAC05980
      IF (A1 .NE. 0 ) R= 1./(1. + ABS(A2/A1))                           DAC05990
      X = IXY(1) + (IXY(3) - IXY(1)) * R                                DAC06000
      Y = IXY(2) + (IXY(4) - IXY(2)) * R                                DAC06010
C  ROUND TO NEAREST INTEGER                                             DAC06020
      IXY(9) = X + SIGN(.5,X)                                           DAC06030
      IXY(10) = Y + SIGN(.5,Y)                                          DAC06040
      IND =1                                                            DAC06050
      RETURN                                                            DAC06060
C  COLINEAR CHECK FOR OVERLAP (COINCIDENCE)                             DAC06070
 2220 IF (ACX *BCX .LT. 0) IND = -1                                     DAC06080
      IF (ADX *BDX .LT. 0) IND= -1                                      DAC06090
      RETURN                                                            DAC06100
      END
```

```
      SUBROUTINE        POLYPT  (N, NX, NY, IND, MINX, MINY )
C
C     ROUTINE  TO  DETERMINE  WHETHER  A  POINT  IS  WITHIN  A  POLYGON
C
C      THE POLYGON BOUNDARY SEGMENTS ARE IN ELIST(I,4), ELIST(I,5)
C     THE NUMBER OF SEGMENTS IS 'N'
C
C    'IND' IS RETURNED:  +1  IF  PT. IS INSIDE
C                         0  IF  PT. IS ON THE BOUNDARY
C                        -1  IF  PT. IS OUTSIDE
C
      DIMENSION  LXY(10),ELIST (2000,5)
      COMMON  ELIST, ZMULT
          INTEGER ELIST
      DOUBLE  PRECISION  ZMULT
      IX =  NX -  MINX + 1
      IY =  NY -  MINY + 1
      IND = 0
      INDX =0
           XX=100000.
      KOUNT =  KOUNT + 1
      LXY(1) = 0
      LXY(2) = IY
      LXY(3) = IX
      LXY(4) = IY
      LXY(9) = 0
      LXY(10)= 0
      IT = 1
      GO TO 2000
C
C
C
C           DRAW A LINE FROM THE POINT  TO THE Y-AXIS AND COUNT THE NUMB
C           OF INTERSECTIONS WITH BOUNDARY SEGMENTS.  ODD IS INSIDE,EVEN
C                 OUTSIDE
C               IF AN INTERSECTION OCCURS AT A NODE
C               CHANGE THE Y COORDINATE AT THE AXIS AND START OVER
C
 1000 IT = IT + 1
      LXY (2) = LXY (2) + IY/10 + 1
      IF (IT.GT.5) RETURN
 2000 INDX = 0
      DO 3000 I = 1,N
      I1 = I + 1
      IF (I.EQ.N) I1 = 1
      LXY (5) = ELIST(I ,4) - MINX + 1
      LXY (6) = ELIST(I ,5) - MINY + 1
      LXY (7) = ELIST(I1,4) - MINX + 1
      LXY (8) = ELIST(I1,5) - MINY + 1
CHECK FOR PT AT A NODE
C
      DO 2110 L=5,7,2
      IF (LXY(L).EQ.IX.AND.LXY(L+1).EQ.IY) RETURN
 2110 CONTINUE
      CALL  INSECT( LXY, INT )
      IF (INT) 1000,3000,2400
C
CHECK FOR PT. ON  BOUNDARY
C
 2400 IF (LXY(9).EQ.IX.AND.LXY(10).EQ.IY) RETURN
C
CHECK FOR INTERSECTION WITH A CORNER
C
      DO 2500 J=6,8,2
      IF( LXY(9)  .NE. LXY(J-1) ) GO TO 2500
      IF (LXY(10).EQ.LXY(J))   GO TO 1000
 2500 CONTINUE
      INDX = INDX + 1
 3000 CONTINUE
      IND = 1
      IF((INDX/2) * 2 .EQ. INDX ) IND = -1
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE REFMT(NBLS,KEY)
      IMPLICIT  INTEGER (A-Z)                                          DAC06130
      DIMENSION ELIST(2000,5)                                          DAC06140
      COMMON ELIST                                                     DAC06150
      KEY1=KEY/10000                                                   DAC06160
      KEY2=MOD(KEY,10000)                                              DAC06170
      WRITE(13,100) KEY1,KEY2                                          DAC06180
  100 FORMAT(1X,2I5)                                                   DAC06190
      XMIN=ELIST(1,4)                                                  DAC06200
      XMAX=ELIST(1,4)                                                  DAC06210
      YMIN=ELIST(1,5)                                                  DAC06220
      YMAX=ELIST(1,5)                                                  DAC06230
      DO 10 J=2,NBLS                                                   DAC06240
      IF(XMIN.GT.ELIST(J,4)) XMIN=ELIST(J,4)                           DAC06250
      IF(XMAX.LT.ELIST(J,4)) XMAX=ELIST(J,4)                           DAC06260
      IF(YMIN.GT.ELIST(J,5)) YMIN=ELIST(J,5)                           DAC06270
      IF(YMAX.LT.ELIST(J,5)) YMAX=ELIST(J,5)                           DAC06280
   10 CONTINUE                                                         DAC06290
      WRITE(13,200) XMIN,YMIN,XMAX,YMAX,NBLS                           DAC06300
  200 FORMAT(4I9,I5)                                                   DAC06310
      M2=0                                                             DAC06320
      MK=NBLS                                                          DAC06330
   20 CONTINUE                                                         DAC06340
      M1=M2+1                                                          DAC06350
      M2=M2+4                                                          DAC06360
      IF(MK.LT.4) M2=NBLS                                              DAC06370
      WRITE(13,300) (ELIST(M,4),ELIST(M,5),M=M1,M2)                    DAC06380
  300 FORMAT(4(2I9))                                                   DAC06390
      MK=MK-4                                                          DAC06400
      IF(MK.GE.0) GO TO 20                                             DAC06410
        RETURN                                                         DAC06420
      END
```

```
              SUBROUTINE  ADJNCY (NBLS, KEY,LIST )
                                                                       DAC01910
      ADJACENCY  LIST                                                  DAC01920
                                                                       DAC01930
      DIMENSION   LIST(2000)                                           DAC01940
      DATA    NOUT3   / 12 /                                           DAC01950
      NEND = NBLS - 1                                                  DAC01960
      DO  1200  I  = 1, NEND                                           DAC01970
      IF( LIST(I) .LT. 0 ) GO TO 1200                                  DAC01980
      NEXT= I + 1                                                      DAC01990
      N    = LIST(I)                                                   DAC02000
                                         UNDUPLICATE  LIST             DAC02010
          DO  1100  J = NEXT, NBLS                                     DAC02020
          IF( LIST(J) .EQ. N ) LIST(J) = -1                            DAC02030
1100      CONTINUE                                                     DAC02040
1200  CONTINUE                                                         DAC02050
C                                        PRINT AND COPY LIST           DAC02060
      WRITE(6, 1300 )  KEY                                             DAC02070
1300  FORMAT(  '0      LIST OF  ADJACENT  AREAS  FOR', I11/)           DAC02080
      DO  1500  I  = 1, NBLS                                           DAC02090
      IF( LIST(I) .LE. 0  ) GO TO 1500                                 DAC02100
      WRITE(6,1400) LIST(I)                                            DAC02110
1400  FORMAT( 50X, I11 )                                               DAC02120
      WRITE(NOUT3,1450 )  KEY, LIST(I)                                 DAC02130
1450  FORMAT( 2I15 )                                                   DAC02140
C                                                                      DAC02150
1500  CONTINUE                                                         DAC02160
      RETURN                                                           DAC02170
      END
```

APPENDIX F

SAMPLE OF AN APL FILE DUMP

```
      (FREAD 13 1)[14;(⍳6),56]
260 268 234 96 47 41 61300000

      (FREAD 12 14)[;(⍳6),56]
 47        36        27        19         7        11  61301000
 49        42        35        12        14         4  61302000
 44        37        28        16         5         8  61303000
 76       115        86        34        15         9  61304000
 44        38        58        15         6         9  61305000

      (FREAD 11 62)[;(⍳6),56]
  4         5         0         0         0         0  61302010
  6         2         1         0         0         0  61302020
  3         4         3         0         2         2  61302030
  7         1         1         2         0         1  61302040
  4         3         4         1         0         0  61302050
  3         2         5         3         6         0  61302060
  7         8         5         0         1         0  61302070
  3         5         6         3         3         0  61302080
  5         3         6         2         0         1  61302090
  2         1         0         0         0         0  61302100
  1         2         1         0         1         0  61302110
  1         2         1         1         1         0  61302140
  1         0         1         0         0         0  61302150
  2         4         1         0         0         0  61302160
```

C.2

TREAD 21 62

61302	61302
706676	1513748
706343	1513758
706384	1514115
706742	1514059
706774	1514432
706408	1514480
706448	1514829
706839	1514775
706448	1514629
706066	1514892
706099	1515234
706448	1514829
706505	1515188
706408	1514480
706384	1514115
705603	1514216
705659	1514583
706408	1514480
706343	1513758
706263	1513406
706343	1513758
705578	1513875
705603	1514216
705229	1514272
706066	1514892
705715	1514941
705293	1515002
705715	1514941
705659	1514583
705262	1514629
705715	1514941
705732	1515289
705578	1513875
705539	1513494
705578	1513875
705182	1513898

ORIGINAL PAGE IS OF POOR QUALITY

```
     FREAD 22 62
276 278 279

     FREAD 23 62
287 289

     FREAD 24 276
 61301      61302
706653  1513359
706676  1513748
706742  1514059
706774  1514432
706839  1514775
706847  1515147

     FREAD 24 277
 61301      61305
707385  1513256
706996  1513303
706653  1513359

     FREAD 24 278
 61302      61303
705149  1513533
705182  1513898
705229  1514272

705262  1514629
705293  1515002
705358  1515345

     FREAD 24 287
 61401      61302
706847  1515147
706505  1515188
706099  1515234

     FREAD 24 289
 61402      61302
705099  1515234
705732  1515289
705358  1515345
```

ORIGINAL PAGE IS
OF POOR QUALITY

FPEAD 33 1

716456	1530243	60100000	1	3
714091	1524391	60200000	2	2
0	0	61190000	3	1
0	0	60300000	4	2
717925	1507257	60400000	5	5
713833	1509448	60500000	6	5
711896	1511701	60600000	7	6
709267	1509064	60700000	8	7
710958	1504727	60800000	9	7
712658	1502652	60900000	10	7
0	0	61000000	11	4
702475	1502625	61100000	12	5
702576	1510172	61200000	13	5
702485	1511139	61300000	14	5
702631	1515698	61400000	15	4
707229	1515084	61500000	16	5
704934	1517626	61610000	17	2
700760	1518152	61620000	18	2
699673	1511396	61700000	19	5
694501	1512450	61800000	20	3
695130	1518123	61900000	21	3
695075	1521767	62000000	22	5
700496	1524652	62100000	23	3
690282	1526167	62200000	24	3
690678	1521710	62300000	25	4
688991	1519979	62400000	26	6
690552	1512631	62500000	27	8

ORIGINAL PAGE IS
OF POOR QUALITY

FPEAD 32 14

706653	1513359	61301000	61	18
705149	1513533	61302000	62	14
703612	1513753	61303000	63	19
702485	1512139	61304000	64	21
705152	1512165	61305000	65	14

FREAD 31 62

706659	1514980	61302010
706617	1514630	61302020
706577	1514272	61302030
706532	1513922	61302040
706455	1513562	61302050
705927	1513635	61302060
705980	1513991	61302070
706012	1514347	61302080
706055	1514709	61302090
706281	1515037	61302100
705902	1515090	61302110
705439	1514426	61302140
705394	1514064	61302150
705363	1513700	61302160

ORIGINAL PAGE IS
OF POOR QUALITY

NASA—JPL—Coml., L.A., Calif